DOI:
10.1016/j.jbi.2016.11.001

Document Version
Peer reviewed version

Anonymizing datasets with demographics and diagnosis codes in the presence of utility constraints

Giorgos Poulis[a], Grigorios Loukides[b], Spiros Skiadopoulos[a], Aris Gkoulalas-Divanis[c]

[a]*Department of Informatics and Telecommunications, University of Peloponnese, Greece.*
[b]*Department of Informatics, King's College London, UK.*
[c]*IBM Watson Health, Cambridge, MA, USA*

Abstract

Publishing data about patients that contain both demographics and diagnosis codes is essential to perform large-scale, low-cost medical studies. However, preserving the privacy and utility of such data is challenging, because it requires: (i) guarding against identity disclosure (re-identification) attacks based on both demographics and diagnosis codes, (ii) ensuring that the anonymized data remain useful in intended analysis tasks, and (iii) minimizing the information loss, incurred by anonymization, to preserve the utility of general analysis tasks that are difficult to determine before data publishing. Existing anonymization approaches are not suitable for being used in this setting, because they cannot satisfy all three requirements. Therefore, in this work, we propose a new approach to deal with this problem. We enforce the requirement (i) by applying (k, k^m)-anonymity, a privacy principle that prevents re-identification from attackers who know the demographics of a patient and up to m of their diagnosis codes, where k and m are tunable parameters. To capture the requirement (ii), we propose the concept of utility constraint for both demographics and diagnosis codes. Utility constraints limit the amount of generalization and are specified by data owners (e.g., the healthcare institution that performs anonymization). We also capture requirement (iii), by employing well-established information loss measures for demographics and for diagnosis codes. To realize our approach, we develop an algorithm that enforces (k, k^m)-anonymity on a dataset containing both demographics and diagnosis codes, in a way that satisfies the specified utility constraints and with minimal information loss, according to the measures. Our experiments with a large dataset containing more than $200,000$ electronic health records show the effectiveness and efficiency of our algorithm.

Keywords: Privacy, Demographics; Diagnosis codes; Utility constraints; Generalization; Suppression

1. Introduction

Healthcare organizations collect increasingly large amounts of data, including clinical trials, Electronic Health Records (EHR), disease registries, and medical imaging databases. In fact, the estimated amount of healthcare data in the world was 0.5 Exabytes ($0.5 \cdot 10^{18}$ bytes) in 2012 and is expected to reach 25 Exabytes by 2020 [66]. Healthcare data are essential for performing large-scale, low-cost analyses [18], which range from Genome-Wide Association Studies (GWAS) to predictive modeling [9, 32] and have the potential to improve medical research and practice. For instance, the study in [14] used more than 350,000 records of the Scandinavian Donations and Transfusions database, along with the donors' and the recipients' health records, to answer whether blood transfusions transmit cancer, and it had a substantial impact on public health policies regarding restrictions placed on blood donors [3, 13]. Another study [75] used an EHR database of over 300,000 records, to learn meaningful comorbidities, which are associated with different stages of Chronic Obstructive Pulmonary Disease (COPD). This study has the potential to improve COPD prognosis, drug development, and clinical trial design.

While the value of analyzing healthcare data is widely recognized, data sharing remains an obstacle for the majority of healthcare providers [17]. In particular, the privacy-preserving sharing of healthcare data beyond authorized recipients (e.g., researchers or employees of the institution that has collected the data) is challenging [15, 16, 25]. This is partly because it cannot be facilitated based on access control and encryption-based methods [59, 65], or by relying solely on policies (e.g., the HIPAA Privacy Rule [57] in the US, the Anonymization Code [56] in the UK, and the Data Protection Directive [58] in the EU). In fact, a major concern in healthcare data publishing is *identity disclosure* (or *re-identification*), an attack in which patients are linked with their records in the published dataset. Identity disclosure can be performed, even when the published dataset is devoid of direct identifiers (e.g., patient phone numbers), due to the availability of external data sources that can be linked to the published dataset, based on demographics [67] or diagnosis codes [47]. For example, Sweeney estimated that 87% of US citizens can be re-identified based on gender, date of birth, and ZIP code, while Golle [27] estimated this percentage as 63%, using newer, Census 2000 data. In addition, Loukides et. al [44] showed that 96% of 2700 patients, who are involved in an NIH-funded GWAS, are uniquely re-identifiable based on their set of diagnosis codes. In response, various methods have been proposed to prevent identity disclosure when publishing

Email addresses: poulis@uop.gr (Giorgos Poulis), gloukides@acm.org (Grigorios Loukides), spiros@uop.gr (Spiros Skiadopoulos), gkoulala@us.ibm.com (Aris Gkoulalas-Divanis)

a dataset that contains demographic attributes (e.g., [16, 39, 64, 79]), or diagnosis codes (e.g., [24, 31, 47, 70]).

In this work, we consider the problem of preventing identity disclosure when we need to publish datasets containing both demographics and diagnosis codes, henceforth termed RT-datasets. Such datasets are used in many applications [61]. Here, we provide some recent examples:

1. The CMS-HCC risk adjustment model [28] uses demographics and diagnoses of health insurance beneficiaries, to predict the health costs of a US health insurance program, called Medicare Advantage. In particular, beneficiaries' demographics (e.g., gender, aged/disabled status, and whether a beneficiary lives in a certain community or close to an institution) and diagnostic data are used to build and update the risk model. The data are provided from hospital inpatient, hospital outpatient, and physicial risk adjustment data.
2. Various epidemiological [11, 26, 62] and cancer research [36] studies are based on data containing demographics and diagnosis codes of patients in New South Wales (NSW), Australia. For example, the study of [36] used the data of women over 45 who are associated with certain diagnosis and procedural codes indicating invasive breast cancer. These data were obtained from the NSW Cancer Registry and from several routinely-collected administrative and self-reported health datasets in NSW, and they were analyzed to find out their predictive power in identifying invasive breast cancer cases.
3. The study of [7] uses a dataset containing demographics and ICD-9 diagnosis codes of patients from various US hospitals, to identify groups of patients that are likely to be diagnosed with diabetes, based on their demographics. In particular, it uses multi-label learning algorithms [74], to estimate the risk that each patient has for being diagnosed with diabetes, based on multiple demographics, such as race, gender, and age group.

These applications use data that are devoid of direct identifiers and thus potentially susceptible to identity disclosure. However, their authors recognize the need for algorithms that anonymize both demographics and diagnosis codes, in order to prevent identity disclosure [7] and increase data availability [36]. Also, publishing RT-datasets is important to support analysis tasks, including case count studies [46, 54], which require accurately counting the number of patients associated with specific demographics and diagnosis codes, predictive modeling, and query answering [61].

However, anonymizing an RT-dataset in a utility-preserving way is a very challenging task. This was acknowledged in [54], which is the first work that studied the general problem of anonymizing an RT-dataset. As explained in Section 2, our work differs from that of [54] in terms of five main dimensions (anonymization principle, data transformation operation, support of utility constraints, information loss criterion, and anonymization algorithm). Specifically, there are three challenges entailed in the anonymization of an RT-dataset in a utility-preserving way. First, identity disclosure cannot be prevented by applying existing methods on demographics and on diagnosis codes separately. This is because an attacker with knowledge of both demographics and diagnosis codes can still re-identify a patient, when the combination of demographics and diagnosis codes of the patient is unique in the anonymized dataset [61]. Specifically, the probability of re-identifying a patient based on such a combination is the reciprocal of the frequency of the combination in the anonymized dataset. Second, data utility must be preserved. This requires constructing an anonymized RT-dataset which allows performing: *(i)* intended analysis tasks with no loss of accuracy, and *(ii)* general analysis tasks, which are difficult to determine before data publishing, with minimum loss of accuracy.

However, the existing methods for anonymizing RT-datasets [54, 61, 68] may substantially reduce the accuracy of intended analysis tasks, or incur excessive information loss, which reduces the accuracy of general analysis tasks. Specifically, the method of [54] does not preserve data *truthfulness*, because it is based on noise addition. That is, it produces synthetic data. Such data are useful for general statistical analysis or mining tasks and can offer strong privacy guarantees [54]. However, the fact that they contain fake information about patients makes them unsuitable for certain applications. For example, they may lead to false alarms in epidemiology [8]. Therefore, our focus is on an anonymization approach that produces truthful data. In addition, the methods of [61] and [68] do not preserve both aspects of data utility; the output of [61] is of little use in intended tasks and that of [68] incurs substantial information loss, which affects the output of general analysis tasks. To illustrate the challenges of anonymizing an RT-dataset, we provide Example 1.

Example 1. *Consider the RT-dataset D in Figure 1a.* Age, Origin *and* Gender *are demographic attributes, and* Disease *is a set containing diagnosis codes, whose description is presented in Figure 2. The dataset in Figure 1b was produced by applying the method of [78] on the demographic attributes and the method of [45] on* Disease. *In particular, the latter dataset satisfies* 2*-anonymity [67] and* 2^2*-anonymity [50, 71], because no record contains a unique combination of demographic values, or a unique combination of two or fewer diagnosis codes (the result of generalizing diagnosis codes is enclosed in* () *and interpreted as any combination of the codes). However, an attacker who knows the demographics and two di-*

agnosis codes of a patient can still re-identify patients. For example, an attacker who knows that Zoe *is a* 30*-year-old* Female *from* Spain*, diagnosed with* 493.2 *(*Chronic obstructive asthma*) and* 494.1 *(*Bronchiectasis with acute exacerbation*), can associate her with the record* 3 *of the dataset in Figure 1b.*

Consider also that the RT-dataset in Figure 1a needs to support a study which requires counting the number of patients who are at most 50 *and are diagnosed with* 494.1*. Applying the method of [61] (respectively, of [68]) produces the anonymized dataset in Figure 1c (respectively, in Figure 1d). However, these datasets cannot support the study, since the number of records corresponding to patients at most* 50 *who are diagnosed with* 494.1 *cannot be accurately determined. This is because the* Age *values of the records* 0 *to* 4 *have been replaced with the range (interval)* [19:51] *in Figure 1c, while* 494.1 *has been generalized together with other diagnosis codes in Figure 1d.*

To address these challenges, our work makes the following specific contributions.

Utility constraints for *RT*-datasets. We investigate how to model and enforce the requirement of supporting case count studies with no accuracy loss. We propose the concept of utility constraint for RT-datasets, building upon previous work on diagnosis codes [46, 50]. A utility constraint represents the least preferred way of generalizing the values of a record in terms of information loss, so that the record does not affect the accuracy of an intended study. For example, the utility constraint

$$u = \{[19{:}50], \mathsf{All}, \mathsf{All}, \{494.1\}\}$$

dictates that:

- a value in Age must be replaced with the range [19:50] or any range contained in [19:50],
- a value in Origin (respectively, in Gender) can be replaced with the most general value All or with any other value in the domain of Origin (respectively, Gender), and
- the diagnosis code 494.1 must be left intact.

Generalizing a record in an RT-dataset as specified by the utility constraint u, ensures that the record remains useful for the study in Example 1. For instance, record 0 of Figure 3 is generalized according to u and does not affect the accuracy of the study, since it certainly corresponds to a patient at most 50 who is diagnosed with 494.1. To support multiple case count studies, we define the concept of utility constraint set.

id	name	Demographics			Diagnosis codes
		Age	Origin	Gender	Disease
0	John	19	France	Male	493.2 494.1 053.20
1	Steve	22	Greece	Male	493.2 494.1 053.71
2	Mary	28	Italy	Female	494.1 053.20
3	Zoe	30	Spain	Female	493.2 494.1 053.71
4	Luke	51	Algeria	Male	493.2
5	Jim	51	Nigeria	Male	493.2 494.1
6	Nicky	47	Ghana	Female	494.1 458.1 458.21
7	Chris	44	Portugal	Male	458.1 494.1

(a) An original RT-dataset D

id	Demographics			Diagnosis codes
	Age	Origin	Gender	Disease
0	[19:22]	Europe	Male	493.2 494.1 (053.20, 053.71)
1	[19:22]	Europe	Male	493.2 494.1 (053.20, 053.71)
2	[28:30]	Europe	Female	494.1 (053.20, 053.71)
3	[28:30]	Europe	Female	493.2 494.1 (053.20, 053.71)
4	51	Africa	Male	493.2
5	51	Africa	Male	493.2 494.1
6	[44:47]	All	All	494.1 (458.1, 458.21)
7	[44:47]	All	All	(458.1, 458.21) 494.1

(b) A version of D that is 2-anonymous with respect to demographics and 2^2-anonymous with respect to diagnosis codes

id	Demographics			Diagnosis codes
	Age	Origin	Gender	Disease
0	[19:51]	All	Male	493.2 494.1 (053.20, 053.71)
1	[19:51]	All	Male	493.2 494.1 (053.20, 053.71)
4	[19:51]	All	Male	493.2
5	[19:51]	All	Male	493.2 494.1
6	[44:47]	All	All	494.1 (458.1, 458.21)
7	[44:47]	All	All	(458.1, 458.21) 494.1
2	[28:30]	Europe	Female	(493.2, 494.1) (053.20, 053.71)
3	[28:30]	Europe	Female	(493.2, 494.1) (053.20, 053.71)

(c) A 2^2-anonymous version of D produced by [61]

id	Demographics			Diagnosis codes
	Age	Origin	Gender	Disease
0	[19:22]	Europe	Male	(493.2, 494.1, 053.20, 053.71)
1	[19:22]	Europe	Male	(493.2, 494.1, 053.20, 053.71)
2	[28:30]	Europe	Female	(493.2, 494.1, 053.20, 053.71)
3	[28:30]	Europe	Female	(493.2, 494.1, 053.20, 053.71)
4	51	Africa	Male	(493.2, 494.1)
5	51	Africa	Male	(493.2, 494.1)
6	[44:47]	All	All	(494.1, 458.1, 458.21)
7	[44:47]	All	All	(494.1, 458.1, 458.21)

(d) A 2-anonymous version of D produced by [68]

Figure 1: An RT-dataset D and several anonymized versions. Attributes id and name are presented only for reference and are not published.

ICD-9 code	Disease
493.2	Chronic obstructive asthma
494.1	Bronchiectasis with acute exacerbation
053.20	Herpes zoster dermatitis of eyelid
053.71	Otitis externa due to herpes zoster
458.1	Chronic hypotension
458.21	Hypotension of hemodialysis

Figure 2: Diseases and their respective ICD-9 codes.

id	Demographics			Diagnosis codes
	Age	Origin	Gender	Disease
0	[19:30]	Europe	All	493.2 494.1 (053.20, 053.71)
1	[19:30]	Europe	All	493.2 494.1 (053.20, 053.71)
2	[19:30]	Europe	All	494.1 (053.20, 053.71)
3	[19:30]	Europe	All	493.2 494.1 (053.20, 053.71)
6	[44:47]	All	All	494.1 (458.1, 458.21)
7	[44:47]	All	All	(458.1, 458.21) 494.1
4	51	Africa	Male	493.2
5	51	Africa	Male	493.2

Figure 3: The $(2, 2^2)$-anonymous dataset produced by applying ART_{UC} to the dataset of Figure 1a.

A utility constraint set contains multiple utility constraints, which can be supported simultaneously.

Anonymization algorithm for *RT*-datasets. We develop ART_{UC}, an algorithm for publishing an RT-dataset that prevents identity disclosure, allows performing intended case count studies accurately, and incurs minimal information loss. Our algorithm thwarts identity disclosure based on (k, k^m)-anonymity [61], a principle that protects from attackers who know the demographics and up to m diagnosis codes of a patient. (k, k^m)-anonymity limits the probability of patient re-identification to at most $\frac{1}{k}$, where k and m are parameters selected by data owners (e.g., the data collecting healthcare institution). ART_{UC} anonymizes the dataset as specified by a given utility constraint set and with low information loss. To cope with the difficulty of minimizing the information loss of the RT-dataset, ART_{UC} processes the demographic attributes and diagnosis codes separately. Specifically, our algorithm partitions the dataset into small clusters with similar values in the demographic attributes and generalizes the values in each cluster. After that, ART_{UC} merges clusters with the same generalized values and anonymizes the diagnosis codes in each cluster. The anonymization employs generalization and suppression and enforces (k, k^m)-anonymity in a utility-preserving way. For

example, ARTUC produced the $(2, 2^2)$-anonymous dataset in Figure 3, when applied to the dataset in Figure 1a. Notice that the probability of re-identification based on the demographics and up to 2 diagnosis codes is at most $\frac{1}{2}$, while the anonymized dataset supports the study of Example 1. In addition, the information loss is lower compared to the datasets produced by methods of [61] and [68] (illustrated in Figures 1c and 1d respectively).

Evaluation of ARTUC. We investigate the effectiveness and efficiency of ARTUC by conducting experiments using a publicly available *RT*-dataset having 5 demographic attributes and over 30,000 electronic health records and an *RT*-dataset having 2 demographic attributes and over 200,000 electronic health records. We also use 9 utility constraint sets, which simulate the requirement of supporting different type of case count studies. Our results show that our algorithm performs anonymization according to the utility constraints and with low information loss. In addition, it takes less than 10 minutes and scales well with the dataset size and anonymization parameters.

The rest of the paper is organized as follows. Section 2 surveys related work. Section 3 presents the fundamental concepts used in this work and the problem statement. Section 4 presents the ARTUC algorithm, and Section 5 the experimental evaluation. Section 6 discusses extensions and limitations of our approach. Last, Section 7 concludes the paper.

2. Related work

In this section, we discuss anonymization methods that are closer to ours (see [15, 52, 25] for surveys). For extensive surveys on anonymization principles and methods for healthcare data, the reader is referred to [17,25,27]. In particular, Section 2.1 discusses algorithms for anonymizing *RT*-datasets, and Section 2.2 provides a brief overview of algorithms for anonymizing demographics and diagnosis codes.

2.1. Anonymization methods for RT-datasets

As mentioned in Section 1, methods for anonymizing an *RT*-dataset have been proposed in [54, 61, 68]. More specifically, the method of Takahashi et al. [68] enforces k-anonymity [67], which requires at least k records to have the same values in all demographic attributes and in the set-valued attribute containing diagnosis codes. To enforce k-anonymity, it employs hierarchy-based generalization [20], effectively replacing each group of values with their closest common ancestor in a given hierarchy. Furthermore, it applies *pre-generalization*,

which generalizes some values in a pre-determined way, selected by data owners. Thus, the method of [68] treats both demographics and diagnosis codes in the same way. As a result, it explores a much smaller space of potential solutions than our algorithm and incurs higher information loss. This is because it: (i) pre-generalizes attribute values, even when they can be generalized with lower information loss (i.e., replaced with more specific values), and (ii) employs the hierarchy-based generalization model for diagnosis codes, which was shown to offer lower data utility than the set-based generalization model we adopt [46].

Poulis et al. [61] proposed a general framework for enforcing (k, k^m)-anonymity on relational and set-valued attributes. In this framework, the $\mathbf{RM}_R$ algorithm stands out, since it anonymizes RT-datasets with low information loss [61]. When applied to an RT-dataset, $\mathbf{RM}_R$ generalizes demographic attributes so the incurred information loss remains lower than a given threshold and performs set-valued generalization to diagnosis codes. However, none of the algorithms in [61] allows performing intended case count studies with no loss of accuracy, as our algorithm, ART_{UC}, does. Furthermore, ART_{UC} outperforms $\mathbf{RM}_R$ in terms of preserving data utility, as shown in our experiments.

Another method for anonymizing an RT-dataset was proposed by Mohammed et al. [54]. The method enforces differential privacy [12], a strong privacy principle, which ensures that the presence or absence of information about an individual in the dataset does not significantly affect the outcome of analysis applied to the dataset. In other words, any inference that an attacker can make about an individual will be (approximately) independent of whether the individual's record is contained in the dataset or not. To enforce differential privacy, the method of [54] constructs a generalized contingency table, which records the counts of all combinations of values, and then adds noise to the counts to satisfy differential privacy. The utility goal of the method is to preserve as much information as possible for building a classifier.

Consequently, the method of [54] differs from our work along five main dimensions:

1. It employs differential privacy, instead of (k, k^m)-anonymity. Thus, it aims to protect the inference of any information about an individual. On the other hand, our method aims to protect from identity disclosure (and can be extended to additionally protect from attribute disclosure, as explained in Section 6).
2. It produces a synthetic dataset, which is the result from adding noise to counts, in order to enforce privacy. Thus, it considers applications where data truthfulness is not necessary. On the other hand, our work employs generalization and/or suppression, which guarantee data truthfulness and make the result of our method suitable for more

applications.

3. It does not consider utility constraints, unlike our work. Thus, it assumes a setting where any change to the values of demographics and/or diagnosis codes is acceptable. By employing utility constraints, our work is applicable to a more general setting.
4. It considers a different information loss criterion than our work. That is, it aims to avoid information loss that harms the task of classification. As acknowledged by the authors of [55], their approach is not developed for "general analysis tasks that focus on attribute values of individual records". On the other hand, our approach adopts general utility measures for demographics and for diagnosis codes, which do not consider a specific task and are able to quantify the uncertainty in interpreting generalized values [46, 79].
5. It employs a fundamentally different algorithm to split the dataset into groups. Specifically, the algorithm employed in [54] works in a top-down fashion. It starts from the coarsest possible partition of the dataset (i.e., all records are contained in the same group) and then iteratively partitions the dataset (i.e., spliting the group into smaller groups), along a single attribute at a time. On the other hand, our algorithm works by first creating groups of records around utility constraints based on demographics and then creating clusters in a bottom-up fashion. Bottom-up anonymization algorithms for data grouping generally explore a larger space of potential solutions that often leads to preserving data utility better ([79]).

2.2. Anonymization methods for demographics and for diagnosis codes

A multitude of algorithms can be used to anonymize demographics. These algorithms can be categorized based on two dimensions: *(i)* their privacy principle, and *(ii)* data transformation strategy, as shown in Figure 1. Observe that these algorithms employ the principles of k-anonymity [67], ℓ-diversity [51], and τ-closeness [42], which protect from different attacks, and transform data using generalization and/or suppression [67], microaggregation [10], or bucketization [76]. In the following, we discuss algorithms that apply k-anonymity using generalization and/or suppression, since they are closer to ours. However, these algorithms are not alternatives to our method, because they cannot prevent identity disclosure in RT-datasets (i.e., protect only the demographics). These algorithms can be classified into three categories, based on the way they work: *(i)* Lattice-based, *(ii)* Partitioning-based, and *(iii)* Clustering-based.

Lattice-based algorithms employ a lattice to encode all the ways of generalizing demographics, and they search the lattice for ways that satisfy k-anonymity with minimum infor-

Algorithm	Principle	Data transformation
Incognito[38]	k-anonymity	Generalization/Suppression
Genetic[34]	k-anonymity	Generalization
Mondrian[39]	k-anonymity	Generalization
TDS[20]	k-anonymity	Generalization
Greedy[78]	k-anonymity	Generalization
Hilb[21]	k-anonymity	Generalization
MDAV[73]	k-anonymity	Microaggregation
CBFS[37]	k-anonymity	Microaggregation
Incognito with ℓ-diversity[51]	ℓ-diversity	Generalization / Suppression
Mondrian with ℓ-diversity[77]	ℓ-diversity	Generalization
Anatomize[76]	ℓ-diversity	Bucketization
Incognito with τ-closeness[42]	τ-closeness	Generalization / Suppression
Mondrian with τ-closeness[43]	τ-closeness	Generalization

Table 1: Algorithms applicable on demographics.

Algorithm	Principle	Data transformation
Apriori[71]	k^m-Anonymity	Generalization
Disassociation[50]	k^m-Anonymity	Disassociation
UGACLIP[46]	Privacy-constrained anonymity	Generalization/Suppression
CBA[45]	Privacy-constrained anonymity	Generalization/Suppression
Greedy[80]	(h, k, p)-Coherence	Suppression
SuppressControl[5]	ρ-Uncertainty	Suppression
TDControl[5]	ρ-Uncertainty	Generalization/Suppression
RBAT[48]	PS-rule based anonymity	Generalization
Tree-based[49]	PS-rule based anonymity	Generalization
Sample-based[49]	PS-rule based anonymity	Generalization
PartialSuppression[35]	ρ-Uncertainty	Suppression

Table 2: Algorithms applicable on diagnosis codes.

mation loss. The search can be performed based on binary-search [63], Apriori-like heuristics [2], and genetic algorithms [34]. Lattice-based algorithms typically incur more information loss than Partitioning [33, 40] and Clustering-based algorithms [4, 23, 41, 78], which work by creating groups of records albeit in different ways. For example, the Partitioning-based algorithm in [40] creates groups using the kd-tree construction mechanism [19], while the Clustering-based algorithm in [78] creates clusters as in bottom-up hierarchical clustering [29]. Generally, Clustering-based algorithms incur a lower amount of information loss compared to the Partitioning-based ones. Our algorithm, ART_{UC}, performs the anonymization of demographics, as the Clustering-based algorithms do.

There are also various algorithms for anonymizing diagnosis codes. These algorithms can be categorized according to their privacy principle and data transformation strategy, as can be seen in Figure 2. These algorithms employ k^m-anonymity [71], privacy-constrained anonymity [46], (h, k, p)-coherence [80], ρ-uncertainty [5], and PS-rule based anonymity [49], and they transform diagnosis codes using generalization and/or suppression. In the following, we discuss algorithms for enforcing k^m-anonymity. These algorithms are closer to ours, because they aim to prevent identity disclosure. However, they are not alternatives to our

method, because they cannot protect RT-datasets (i.e., protect only the diagnosis codes). Terrovitis et al. [71] proposed an algorithm, called Apriori Anonymization (AA), which works iteratively in a bottom-up fashion. In each iteration, it identifies all sets of diagnosis codes of a certain size i (initially $i = 1$) and applies generalization to make the dataset k^i-anonymous. Then, it considers all sets of diagnosis codes of size $i + 1$ and repeats the same process. When $i = m$ the process ends and a k^m-anonymous dataset is produced. However, AA does not allow performing intended case count studies with no loss of accuracy [46]. In response, Loukides et al. [50], proposed a method for enforcing k^m-anonymity by *disassociation* [70], an operation that splits the diagnosis codes in records into non-overlapping subsets. The method first creates clusters of records that have semantically similar diagnosis codes and then applies disassociation to the records of each cluster to enforce k^m-anonymity. However, the method of [50] does not preserve data truthfulness. Loukides et al. also proposed the UGACLIP and CBA algorithms in [46] and [45], respectively. These algorithms apply generalization and suppression, and they aim to prevent re-identification based on specific sets of diagnosis codes that are provided as input. They apply generalization and suppression in a way that preserves specific associations between diagnosis codes, which are modeled as utility constraints. CBA is more effective than UGACLIP in terms of preserving the specified associations and in terms of incurring low information loss. In ART_{UC}, we use an adaptation of CBA to anonymize diagnosis codes.

3. Background and problem statement

This section discusses the fundamental concepts that are used in our approach, and it provides the problem statement.

3.1. RT-datasets and their protection against identity disclosure

We consider a dataset in which every record corresponds to a distinct patient and contains a number of *demographic* attributes, as well as a set-valued attribute containing *diagnosis codes*. Such a dataset is referred to as an RT-dataset. Without loss of generality, we assume that the first l attributes in an RT-dataset correspond to the demographic attributes, $A_1, \ldots, A_l$, and the last attribute, A_{l+1}, is a set containing the diagnosis codes of the patient. Extensions to this modeling dealing with multiple set-valued attributes are discussed in Section 6. Each demographic attribute takes values from a different domain, which can be numerical (e.g., for Age) or categorical (e.g., for Gender). Following [67], we also assume that

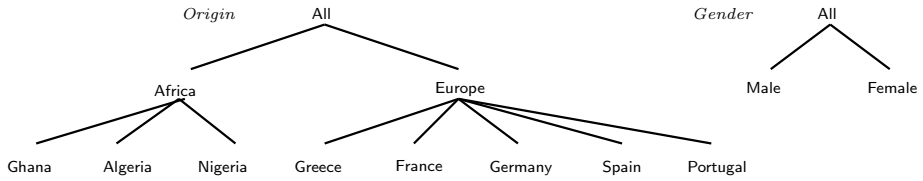


Figure 4: Hierarchies for the dataset of Figure 1a

there is an underlying *hierarchy*[1], for each categorical demographic attribute. On the other hand, the attribute A_{l+1} is a subset of the set of ICD-9 codes [46]). In addition, following existing works [31, 44, 46, 50], we consider ICD-9 codes. If some records contain ICD-10 codes, they can be easily mapped to ICD-9 codes, using *General Equivalence Mappings*, as explained in [6].

Let us now explain how identity disclosure can be performed by an attacker on an *RT*-dataset. We assume that the attacker knows all the values of a patient in each demographic attribute, $A_1, \ldots, A_l$, and up to m diagnosis codes of the attribute A_{l+1}. Knowledge about demographic attributes can be obtained from linking external datasets (e.g., voter lists with hospital discharge summaries), as explained in [67]. In addition, knowledge about diagnosis codes may be solicited from external data sources, including the electronic health record system, as explained in [46], or be background knowledge. The parameter m is an integer, which is set by data owners. The values of m range from 0 to the number of distinct diagnosis codes contained in the dataset. The minimum value does not prevent identity disclosure based on diagnosis codes, while the maximum value protects from attackers with knowledge about any combination of diagnosis codes that a patient may have. Commonly, data owners set m to a small constant [50], since it is unlikely for attackers to know many diagnosis codes for a certain individual.

To prevent identity disclosure based on the aforementioned knowledge, we employ (k, k^m)-anonymity [61], which is defined below.

Definition 1. *An RT-dataset is (k, k^m)-anonymous, when an attacker who knows:*

1. *any combination of the demographic attributes of a patient, and*
2. *any combination of at most m diagnosis codes of the patient,*

cannot use this knowledge to distinguish a record from at least other $k-1$ records in the dataset, where k and m are anonymization parameters *specified by data owners.*

[1] A hierarchy is a tree structure whose leaves represent the original values in an attribute and internal nodes represent more abstract values that summarize their descendants in the tree. See, for example, Figure 4.

(k, k^m)-anonymity limits the probability of identity disclosure, based on the knowledge specified in Items 1 and 2 of Definition 1, to at most $1/k$. Clearly, larger values of k and m achieve higher privacy protection. For example, the dataset in Figure 3 is $(2, 2^2)$-anonymous, because an attacker who knows the values of a patient in the set of demographic attributes {Age, Origin, Gender}, as well as up to 2 diagnosis codes, cannot re-identify the patient with probability larger than $\frac{1}{2}$.

Observe that (k, k^m)-anonymity provides the same protection as k-anonymity [64], for demographic attributes, and as k^m-anonymity [72], for the diagnosis codes attribute. However, *the inverse does not hold*. Specifically, an RT-dataset may be k and k^m but not (k, k^m)-anonymous. For instance, the dataset of Figure 1b is both 2-anonymous and 2^2-anonymous, but it is *not* $(2, 2^2)$-anonymous. Therefore, an attacker who knows the demographics and up to 2 diagnosis codes of a patient may re-identify the patient, as discussed in Example 1. On the contrary, this is not possible when the $(2, 2^2)$-anonymous dataset of Figure 3 is used instead, because at least two records could belong to the patient.

3.2. Generalization, suppression and data utility quantification

To enforce (k, k^m)-anonymity, we may employ *generalization* and/or *suppression*. Generalization replaces a value with a more general value, while suppression deletes a value. Both operations have been applied to anonymize demographics and/or diagnosis codes (see Section 2). Suppression is a more drastic operation than generalization, but it is useful to avoid generalizations of diagnosis codes that have very low data utility [46]. For generalization, we apply a *local recoding* [39] generalization model that is inspired by [61]. Given a cluster of records C, our model replaces the values in each attribute of the records in the cluster, as explained below.

Definition 2. *Given a cluster C, generalization replaces the values of all records in C, in each attribute A, as follows:*

If A is a numerical demographic attribute, *the values are replaced by the range (interval) comprised of the minimum and maximum of these values.*

If A is a categorical demographic attribute, *the values are replaced by their* closest common ancestor *in the hierarchy of A. The common closest ancestor is the node that belongs to every path from a value to the root of the hierarchy and is as far as possible from the root.*

If A is the set-valued attribute, *each value (set of diagnosis codes) of a record in C is replaced by one or more sets of diagnosis codes.*

Thus, the generalization model replaces the value of a record in the cluster, with respect to an attribute, with a node in the hierarchy of the attribute, or with a range, or with set(s) of diagnosis codes, depending on the type of the attribute. For brevity, we refer to the result of applying generalization to a value in a demographic attribute (i.e., a node in the hierarchy, or range) as *generalized value*, and to a set of diagnosis codes that is created by applying generalization to the value in the set-valued attribute as *generalized diagnosis code*. A *generalized value* is interpreted as a single node in the hierarchy or a range, whereas a generalized diagnosis code is interpreted as *any* subset of the diagnosis codes that it has replaced.

Note that there is a single way to replace the values in a cluster with respect to a demographic attribute, whereas there are multiple ways to perform the same for the diagnosis codes in the set-valued attribute. That is, our generalization model does not specify exactly which generalized diagnosis codes will be created. This allows our anonymization algorithm to select the best way to create generalized diagnosis codes, among a large number of possible alternatives (exponential in the number of distinct codes in the cluster), which is important to preserve data utility. Example 2 below illustrates our generalization model.

Example 2. *Consider a cluster containing the records* 6 *and* 7 *of the RT-dataset in Figure 1a and the result of applying the generalization model to the cluster, shown in Figure 5. Observe that the values* {44,47} *in the numerical demographic attribute* Age *have been replaced by the range* [44:47]. *This implies that the records in the cluster may have any value from* 44 *to* 47 *in* Age. *In addition, the values* {Ghana, Portugal} *in the categorical demographic attribute* Origin *have been replaced by their closest common ancestor* All *in the hierarchy for* Origin, *which is shown in Figure 4. Similarly, the values* {Male, Female} *in the categorical demographic attribute* Gender *have been replaced by their closest common ancestor* All *in the hierarchy for* Gender *of Figure 4. Moreover, the diagnosis codes have been replaced by the generalized diagnosis codes* (494.1)[2], *and* (458.1,458.21). *The generalized diagnosis code* (494.1) *is interpreted as* Chronic hypotension, *and the generalized diagnosis code* (458.1,458.21) *is interpreted as* Chronic hypotension *and/or* Hypotension of hemodialysis.

Clearly, our generalization model can enforce (k, k^m)-anonymity on an RT-dataset, when the following conditions hold:

Condition 1: Each cluster contains at least k records with the same generalized value, in each demographic attribute.

[2]For clarity in all figures, we omit () from generalized diagnosis codes comprised of one diagnosis code.

id	Demographics			Diagnosis codes
	Age	Origin	Gender	Disease
6	[44:47]	All	All	494.1 (458.1, 458.21)
7	[44:47]	All	All	(458.1, 458.21) 494.1

Figure 5: Example of applying generalization to the cluster containing the records $\{6, 7\}$ of the dataset of Figure 1a.

Condition 2: Each combination of m or fewer diagnosis codes appears in at least k records of the cluster (a diagnosis code appears in a generalized diagnosis codes when it has been replaced by it after generalization).

Condition 1 (respectively, Condition 2) ensures that the attacker cannot use their knowledge about the demographics (respectively, the diagnosis codes) of the patient in Definition 1 to re-identify the patient with probability that exceeds $\frac{1}{k}$.

There are many different ways to anonymize an RT-dataset, using generalization and/or suppression, which do not offer the same utility. Typically, the utility of an anonymized dataset in general analysis tasks is measured based on: *(a)* the amount of information loss incurred by anonymization, and *(b)* the accuracy of answering aggregate queries using the anonymized dataset.

To measure the amount of information loss, there are various utility measures that are applicable to demographics [4, 22, 39, 79] or diagnosis codes [30, 47, 72]. For demographics, we use the *NCP* (Normalized Certainty Penalty) measure [79], due to its flexibility in dealing with both numerical and categorical demographic attributes and its ability to accurately quantify the uncertainty in interpreting generalized values [79]. The following definitions explain the *NCP* measure.

Definition 3. *Given a generalized value $\tilde{u}$ in a demographic attribute A, the NCP of $\tilde{u}$ is defined as:*

$$NCP_A(\tilde{v}) = \begin{cases} 0, |\tilde{v}| = 1 \\ |\tilde{v}|/|A|, \textit{otherwise} \end{cases},$$

where $\tilde{v}$ and $|A|$ are defined as follows.

If A is numerical, *$\tilde{v}$ is the length of the range $\tilde{v}$ and $|A|$ is the domain size of A.*

If A is categorical, *$\tilde{v}$ is the number of leaves of the subtree rooted at $\tilde{v}$ in the hierarchy of A and $|A|$ is the number of leaves in the hierarchy of A.*

Definition 4. *The NCP of a record r, a cluster C, and an RT-dataset D, is defined as:*

$$NCP(r) = \sum_{i \in [1,l]} w_i \cdot NCP_{A_i}(r[A_i]), \; NCP(C) = \sum_{r \in C} NCP(r) \text{ and } NCP(D) = \frac{\sum_{r \in D} NCP(r)}{|D|}$$

respectively, where $w_i \in [0,1]$ *is a weight that measures the importance of a demographic attribute* A_i, $i \in [1,l]$ *and is specified by data owners,* $r[A_i]$ *denotes the projection of the record* r *on the attribute* A_i, *and* $|D|$ *is the size (number of records) of the dataset.*

The *NCP* measure takes values in the $[0,1]$ range. Lower values of *NCP* indicate lower data distortion and are preferable. Example 3 below illustrates the computation of the *NCP* measure.

Example 3. *Consider the record* 6 *in the anonymized RT-dataset of Figure 3. The NCP of the generalized value* [44:47] *in the numerical demographic attribute* Age *is* $\frac{47-44}{51-19} = 0.094$. *In addition, the NCP of the generalized value* All *in the categorical demographic attribute* Origin *is* $\frac{8}{8}$, *because the number of leaves of the subtree rooted at* All, *in the hierarchy of Figure 4, is* 8, *and the number of leaves in the same hierarchy is also* 8. *Similarly, the NCP of the generalized value* All *in the categorical demographic attribute* Gender *is* $\frac{2}{2}$. *Thus, assuming a weight* $\frac{1}{3}$ *for each demographic attribute, the NCP of the record* 6 *is* $\frac{1}{3}\cdot 0.097 + \frac{1}{3}\cdot 1 + \frac{1}{3}\cdot 1 = 0.7$. *The NCP of the record* 7 *is also* 0.7, *and thus the NCP of the cluster* {6,7} *is* $2 \cdot 0.7 = 1.4$. *Last, the NCP value of the anonymized dataset in Figure 3 is* $\frac{4.28}{8}$, *where* 4.28 *is the sum of the NCP values of all records in the dataset.*

For diagnosis codes, we use the *UL* (Utility Loss) [46] measure, which captures the information loss caused by generalization and suppression and it is suitable to use together with generalization models that replace diagnosis codes with sets [46], such as our generalization model (see Definition 2). The following definitions explain the *UL* measure.

Definition 5. *Given a generalized diagnosis code* $\tilde{u}$, *which has replaced* $|\tilde{u}|$ *diagnosis codes in the RT-dataset, and a weight* $w(\tilde{u}) \in [0,1]$, *which reflects the importance of the diagnosis codes replaced by* $\tilde{u}$ *and is specified by data owners, the UL of* $\tilde{u}$ *is defined as:*

$$UL(\tilde{u}) = (2^{|\tilde{u}|} - 1) \cdot w(\tilde{u}).$$

Definition 6. *The UL of a record* r, *a cluster* C, *and an RT-dataset* D, *is defined as:*

$$UL(r) = \frac{\sum_{\tilde{u} \in r} UL(\tilde{u})}{2^{\sigma(r)} - 1} + s(r), \quad UL(C) = \sum_{r \in C} UL(r) \quad \text{and} \quad UL(D) = \frac{\sum_{r \in D} UL(r)}{|D|}$$

respectively, where $\sigma(r)$ *computes the total number of diagnosis codes that appear in generalized diagnosis codes in* r, $s(r)$ *computes the number of diagnosis codes that have been suppressed from* r, *and* $|D|$ *denotes the size (number of records) of the dataset.*

The *UL* measure takes values in the $[0,1]$ range. Lower values of *UL* indicate lower data distortion and are preferable. Example 4 below illustrates the computation of the *UL* measure.

Example 4. *Consider the record* 6 *in the anonymized RT-dataset of Figure 3 and assume that the weight of each generalized diagnosis code is* 1*. The UL of the generalized diagnosis code* (494.1) *is* $(2^0-1)\cdot 1 = 0$ *and that of* (458.1,458.21) *is* $(2^2-1)\cdot 1 = 3$*. This reflects that no information loss is incurred when* 494.1 *is generalized to* (494.1)*, since the generalized diagnosis code is interpreted as the diagnosis code* 494.1*. Furthermore, it reflects that* (458.1,458.21) *incurs more information loss than* (494.1)*, which is expected because* (458.1,485.21) *is interpreted as* 458.1 *and/or* 458.21*. In addition,* 3 *diagnosis codes (i.e.,* 494.1*,* 458.1*, and* 458.21*) appear in the generalized diagnosis codes in the record* 6*, and no diagnosis codes have been suppressed. Thus, the UL of the record* 6 *is* $\frac{0+3}{2^3-1} + 0 = 0.43$*. Similarly, the UL of the record* 7 *is* 0.43*, and thus the UL of the cluster* $\{6, 7\}$ *is* $2 \cdot 0.43 = 0.86$*. The UL of the dataset in Figure 3 is* $(3 \cdot \frac{2^2-1}{15} + \frac{2^2-1}{7} + 2 \cdot \frac{2^2-1}{4} + 0 + 1)/8 = 0.441$.

To measure the accuracy of answering aggregate queries using anonymized data, we employ the *ARE* (Average Relative Error) measure [39]. In particular, we consider COUNT queries that model case count studies and have the following SQL-like form:

```
SELECT COUNT(*)
FROM 𝒟 (or D)
WHERE R and T are supported by a record r in 𝒟 (or in D)
```

where R (respectively, T) is a set of values in demographic attributes (respectively, a set of diagnosis codes), and D and $\mathcal{D}$ is the RT-dataset before and after anonymization, respectively. That is, a query asks for the number of patients with certain demographics that are diagnosed with a set of one or more diagnosis codes. A similar setting involving only diagnosis codes was considered in [46, 47, 70].

Lower *ARE* values indicate more accurate queries answers and are preferable. The following definition explains how *ARE* can be computed.

Definition 7. *For a workload* $\mathcal{W}$ *of* COUNT *queries* $q_1, \ldots, q_n$*, and for two functions* C_A *and* C_O*, which count the number of records answering a query* $q_i, i \in [1, n]$ *on the anonymized dataset* $\mathcal{D}$ *and on the original dataset* D *respectively, the ARE measure is defined as:*

$$ARE(\mathcal{W}) = avg_{i\in[1,n]} \frac{|C_A(q_i) - C_O(q_i)|}{C_O(q_i)}$$

Thus, *ARE* is computed as the mean error of answering all queries in the query workload $\mathcal{W}$. Clearly, a zero *ARE* implies that the anonymized dataset $\mathcal{D}$ is as useful as the original dataset D in answering the queries in $\mathcal{W}$, and low values in ARE are preferred. Note that anonymized datasets that have low *ARE* values are useful not only for answering COUNT queries but also for various analysis tasks that are based on such queries, such as frequent itemset mining and classification [40].

$u = \{[40{:}50], \text{Europe}, \text{All}, \{053.20, 053.70\}\}$

Table 3: An example of a utility constraint.

3.3. Utility constraints for RT-datasets

Minimizing a data utility measure, such as *NCP* or ARE, is important. However, it does not guarantee that the anonymized dataset will be useful in intended analysis tasks, as explained in Section 1. This has been recognized by prior works on anonymizing diagnosis codes [46, 50], which proposed the concept of *utility constraints* (see Section 2.2).

In our work, we extend this concept to *RT*-datasets. Intuitively, a utility constraint models the least preferable way in which a record can be generalized, in the sense that applying more generalization yields the record useless for performing the intended task (i.e., the task is performed with lower accuracy after anonymization).

Definition 8. *A utility constraint u is an ordered set $\{v_{A_1}, \ldots, v_{A_l}, v_{A_{l+1}}\}$ where v_{A_i}, $i \in [1, l+1]$ is defined as follows:*

If A_i is a numerical demographic attribute, *v_{A_i} is a range of values of A_i.*

If A_i is a categorical demographic attribute, *v_{A_i} is a node in the hierarchy of A_i.*

If A_i is the set-valued attribute, *v_{A_i} is a set of diagnosis codes.*

An example of a utility constraint is shown in Table 3. This utility constraint is comprised of the range [40:50] in the numerical demographic attribute Age, the nodes Europe and All in the hierarchy of the categorical demographic attribute Origin and Gender, respectively (see Figure 4), and the set of the diagnosis codes 053.20 and 053.70, in the set-valued attribute.

Our definition of a utility constraint does not pose any restriction in the choice of hierarchy nodes, ranges, or set of diagnosis codes. However, such restrictions are necessary when there are multiple utility constraints, to avoid conflicts between utility constraints which can render the anonymized data useless for the intended task. For example, consider the utility constraint u in Table 3 and another utility constraint $u' = \{[19{:}51], \text{Germany}, \text{Female}, \{053.20\}\}$. Clearly, there is a conflict between these two utility constraints, because u states that the least acceptable way to generalize a value in Age is to replace it with [40:50], whereas u' states that replacing the value with [19:51] (i.e., further generalizing the value) is still acceptable. To avoid such conflicts, we require the set of utility constraints to satisfy *all* of the following requirements.

For each numerical demographic attribute, any two utility constraints should have the same range or two disjoint ranges.

For each categorical demographic attribute, any two utility constraints should have the same node or two nodes that have no common leaf-level descendant in the hierarchy of the attribute.

For the set-valued attribute, any two utility constraints should have the same set of diagnosis codes or two disjoint sets of diagnosis codes.

In the following, we assume that these requirements are satisfied by the specified utility constraints. In addition, we require the specified utility constraints to *cover* each value in an RT-dataset. More specifically, we require:

For each numerical demographic attribute A_i**,** each respective value to be contained in the element (range) v_{A_i} of a utility constraint.

For each categorical demographic attribute A_i**,** each respective value to be contained in the subtree of the hierarchy of A_i that is rooted at the element v_{A_i} of a utility constraint.

For the set-valued attribute A_{l+1}**,** each diagnosis code to be contained in the element $v_{A_{l+1}}$ of a utility constraint.

When the specified utility constraints *cover* each value in an RT-dataset, the value can be generalized by our algorithm, in a way that guarantees data utility for the intended analysis task, as will be explained later. When there are no specific requirements for certain attribute values, a utility constraint with a "coarse" element (e.g., All) can be specified for these values. For example, if a case count study does not require distinguishing patients according to their Gender, all utility constraints can have the node All in their element for Gender. In the following, we assume that the specified utility constraints cover each value in the given RT-dataset.

The set that contains all the utility constraints is defined as the *utility constraint set* and is denoted with $\mathcal{U}$. We now explain when an anonymized RT-dataset satisfies a given utility constraint set.

Definition 9. *An anonymized dataset $\mathcal{D}$ satisfies a given utility constraint set $\mathcal{U}$ when, for each record r of $\mathcal{D}$, there is a utility constraint u in $\mathcal{U}$ such that the following conditions hold:*

$u_1 = \{[19{:}50], \mathsf{All}, \mathsf{All}, \{458.1, 458.21\}\}$
$u_2 = \{[19{:}50], \mathsf{All}, \mathsf{All}, \{053.20, 053.70\}\}$
$u_3 = \{[19{:}50], \mathsf{All}, \mathsf{All}, \{494.1\}$
$u_4 = \{[51], \mathsf{Africa}, \mathsf{Male}, \{493.2\}\}$

Table 4: A utility constraint set.

For each numerical demographic attribute A_i**,** *the generalized value of* r *in* A_i*, is contained in the element* v_{A_i} *of* u*.*

For each categorical demographic attribute A_i**,** *the generalized value of* r *of* A_i *is contained in the subtree that is rooted at* v_{A_i} *in the hierarchy of* A_i*, where* v_{A_i} *is the element of* u*.*

For the set-valued attribute A_{l+1}**,** *and for each generalized diagnosis code of* r*, the diagnosis codes that appear in the generalized diagnosis code are contained in the element* $v_{A_{l+1}}$ *of* u*, or of another utility constraint in* $\mathcal{U}$*.*

Thus, when a utility constraint set is satisfied, generalization does not reduce the utility of the RT-dataset for performing the intended analysis task, because no generalized value or generalized diagnosis code exceeds the maximum allowable level of generalization that is dictated by the specified utility constraints.

Example 5 below illustrates when a utility constraint set is satisfied.

Example 5. *The anonymized RT-dataset of Figure 3 satisfies the utility constraint set* $\mathcal{U} = \{u_1, u_2, u_3, u_4\}$*, shown in Table 4, because for each record of the anonymized dataset, its generalized values in* Age*,* Origin*, and* Gender *are contained in elements of the same utility constraint and the diagnosis codes that appear in each generalized diagnosis code are contained in a single utility constraint. For example, the generalized values* [19:30]*,* Europe*, and* All *of record* 1 *are all contained in* u_1 *(or equivalently in* u_2 *or* u_3*, since these two utility constraints have the same elements in* {Age,Origin,Gender} *with* u_1*), and the diagnosis codes* 053.20 *and* 053.71*, which appear in the generalized diagnosis code* (053.20,053.71) *are contained in the same utility constraint* u_2*.*

Note that Definition 9 does not capture the impact of suppressing diagnosis codes on data utility. Thus, the utility constraint set in Example 5 is satisfied when 494.1 has been suppressed from record 5. To control the impact of suppression on diagnosis codes, our problem restricts the allowable number of diagnosis codes that can be suppressed, as we will discuss in Section 3.4.

3.4. Problem statement

We now formally define the problem that we aim to solve in this work, as follows.

Problem 1. *Given an RT-dataset D, a utility constraint set $\mathcal{U}$, parameters k and m, and thresholds δ and ϵ, construct a (k, k^m)-anonymous version $\mathcal{D}$ of D, such that all following requirements hold:*

Requirement 1: *$NCP(\mathcal{D}) \leq \delta$.*

Requirement 2: *$UL(\mathcal{D})$ is minimum.*

Requirement 3: *$\mathcal{D}$ satisfies the utility constraint set $\mathcal{U}$.*

Requirement 4: *At most ϵ diagnosis codes have been suppressed to construct $\mathcal{D}$.*

Solving Problem 1 is far from trivial for three reasons. First, the problem is computationally difficult (NP-hard), even when the utility constraint set $\mathcal{U}$ does not limit the generalization of values (i.e., it contains a utility constraint with the element All for each demographic attribute and the set of all ICD-9 codes for the set-valued attribute) [61]. Second, the requirement to satisfy the utility constraint set, which we introduce in this paper, calls for new methods that are able to take into account the allowable ways of generalizing values. Third, minimizing the information loss in both demographic attributes and diagnosis codes together is difficult. Intuitively, this is because minimizing *NCP* calls for creating small groups of records, whereas minimizing *UL* calls for creating large groups of records, as explained in [61]. Thus, there is a trade-off between the information loss incurred by generalization on demographic attributes and on diagnosis codes. To control this trade-off, we use the thresholds δ and ϵ, which are set by data owners.

Currently, there are no algorithms for solving Problem 1.

4. Anonymization methodology

To address Problem 1, we propose the ART$_{UC}$ algorithm. In the following, we explain the operation of ART$_{UC}$ (Section 4.1) and illustrate its use with an example (Section 4.2). In addition, we propose an adaptation of ART$_{UC}$ to help the data owner decide the maximum allowable information loss in the demographic attributes and the maximum allowable suppressed diagnosis codes (Section 4.3). The adaptation is based on progressively relaxing Requirements 1 and 4 of Problem 1.

4.1. The ART$_{UC}$ *algorithm*

The ART$_{UC}$ algorithm works in four phases:

1. **Record grouping:** In this phase, groups of records are created. Each group contains all records whose values in the set of demographic attributes are contained in the same utility constraint.

2. **Cluster formation:** In this phase, each group is further split into clusters, which become k-anonymous, with respect to the demographic attributes, with minimal NCP.

3. **Cluster merging:** The clusters created in the cluster formation phase are merged. The objective is to create clusters whose diagnosis codes can be anonymized with low information loss, without violating Requirement 1 of Problem 1.

4. **(k, k^m)-anonymization:** Each cluster becomes (k, k^m)-anonymous, by generalizing diagnosis codes. The generalization is performed in a way that aims to satisfy the Requirements 2, 3, and 4 of Problem 1.

After that, the anonymized dataset, which is comprised of the (k, k^m)-anonymous clusters, is returned, and the algorithms terminates. We now describe the pseudocode of ARTU_{UC} in more detail.

Record grouping. The record grouping phase is performed in Step 1. In this phase, the records of the RT-dataset are placed into different groups, and each record belongs to one group that is formed around a different utility constraint. That is, the values of all records in the group in a numerical (respectively, categorical) demographic attribute A_i are contained in the element v_{A_i} (respectively, in the subtree of the hierarchy of A_i that is rooted at the element v_{A_i}) of the utility constraint, for each attribute A_i, $i \in [1, l]$. The specification of the utility constraint set (see the requirements for the specified utility constraints in Section 3.3) guarantees that each record belongs to at least one group. Typically, the created groups contain more than k records.

This is reasonable since k is a small constant in practice. In the extreme case where a group has fewer than k records, generalization alone cannot enforce (k, k^m)-anonymity. Thus, to achieve this goal, the algorithm suppresses the records of the group (Step 2). In our experimental analysis (Section 5), we demonstrate that such suppressions are extremely rare for a wide range of parameter settings and utility constraint sets. Specifically, ARTU_{UC} executed Step 2 only for $k = 100$ (20 times larger than the typical value) and for only one utility constraint. As a result, fewer than 0.01% of the input records were suppressed.

Cluster formation. The cluster formation phase is performed in Steps 3 to 16. In each iteration, a different group $G \in \mathcal{G}$ is considered and is divided into clusters, each containing

Algorithm: ART$_{UC}$

Input: An RT-dataset D, a utility constraint set $\mathcal{U}$, and anonymization parameters k, m, δ and ϵ.

Output: A (k, k^m)-anonymous RT-dataset $\mathcal{D}$ that corresponds to D, satisfies the Requirements 1, 2, and 4 of Problem 1, and has minimal UL.

`// Record grouping`

1 Group the records of D into a set $\mathcal{G}$ of $|\mathcal{U}|$ groups such that the records of each group are contained in the same utility constraint of $\mathcal{U}$.

2 Remove all groups from $\mathcal{G}$ that have less than k records.

`// Cluster formation`

3 Let $\mathcal{C}$ be an empty set of clusters.

4 **foreach** *group* $G \in \mathcal{G}$ **do** `// Form k-sized clusters`

5 **while** *group* G *has more than* k *records* **do**

6 Move a random record from group G into a new cluster C.

7 **while** C *has fewer than* k *records* **do**

8 Find the record $r_{NCP} \in G$ that minimizes the NCP of $C \cup \{r_{NCP}\}$ when generalized with Definition 2.

9 Move record r_{NCP} from group G into cluster C.

10 Add cluster C into the set of clusters $\mathcal{C}$.

11 **foreach** *group* $G \in \mathcal{G}$ **do** `// Consider the groups with records that are not assigned in clusters`

12 **foreach** *record* r *in group* G **do**

13 Find the cluster C_{NCP} in $\mathcal{C}$ that minimizes the NCP of $C_{NCP} \cup \{r\}$ when generalized with Definition 2.

14 Move r from group G into cluster C_{NCP}.

15 Apply the generalization of Definition 2 to each cluster in the set of clusters $\mathcal{C}$.

16 **if** $NCP(\mathcal{C}) > \delta$ **then return** false ;

`// Cluster merging`

17 Merge all clusters of $\mathcal{C}$ with the same generalized values in all the demographic attributes.

18 **repeat**

19 Select the cluster $C_{UL} \in \mathcal{C}$ with minimum UL when generalized with our adaptation of the CBA algorithm.

20 Search for a cluster $C'_{UL} \in \mathcal{C} - C_{UL}$ such that (a) $C_{UL} \cup C'_{UL}$ has minimum UL (when generalized with our adaptation of the CBA algorithm), and (b) NCP of $\mathcal{C} - C_{UL} - C'_{UL} \cup \{C_{UL} \cup C'_{UL}\}$ is at most δ (when generalized with Definition 2).

21 **if** C'_{UL} *exists* **then** Remove from $\mathcal{C}$ clusters C_{UL} and C'_{UL} and add cluster $C_{UL} \cup C'_{UL}$.;

22 **until** *the set of clusters* $\mathcal{C}$ *does not change*;

`// (k, k^m)-anonymization`

23 Let $\mathcal{D}$ be an empty set of clusters that will store the anonymized result.

24 **foreach** *cluster* C *in* $\mathcal{C}$ **do**

25 Anonymize C using our adaptation of the CBA algorithm and add to $\mathcal{D}$ result.

26 **if** *the total number of suppressed diagnosis codes exceeds* ϵ **then return** false ;

27 **return** $\mathcal{D}$

Algorithm 1: ART$_{UC}$

k records. This is performed by a heuristic that adds into the cluster the record that incurs the minimum increase to the NCP value of the cluster, after generalization, until the cluster contains k records (Steps 6 to 10). Note that NCP is measured for the cluster, after the generalization of all its values in the demographic attributes. The generalization is performed by applying Definition 2 to each demographic attribute. Note also that the NCP value of the cluster cannot decrease, after the record addition (see Definition 3). Each created cluster is also saved into the set of clusters $\mathcal{C}$ (Step 10).

Next, the algorithm considers the records that have not been added into clusters (Steps

11 to 14). Clearly, in every group, there can be at most $k-1$ such records; if there were more, they would have formed a cluster. Each of these records is moved into the cluster that fits best, in the sense that its addition incurs the lowest increase in the *NCP* of the cluster. Then, the demographics of each cluster are generalized using Definition 2 (Steps 15). Following, the algorithm checks if Requirement 1 of Problem 1 is satisfied (Step 16). Specifically, if the *NCP* of the resultant clustering $\mathcal{C}$ exceeds the parameter δ, then the algorithm terminates, since an acceptable solution in terms of the information loss with respect to the demographic attributes cannot be found.

Cluster merging. The cluster merging phase is performed in Steps 17 to 22. First, ARTUC combines clusters which have the same generalized values in all demographic attributes (Step 17). Clearly, the resultant clusters are still k-anonymous, and the *NCP* of the set of clusters $\mathcal{C}$ does not change (see Definition 4). However, the merged clusters offer more room for minimizing the information loss of the set-valued attribute. This is because they contain more diagnosis codes and are more likely to become k^m-anonymous with respect to the diagnosis codes, with lower information loss according *UL*.

Subsequently, ARTUC merges together clusters with semantically close diagnosis codes aiming at minimizing *UL* (Steps 18 to 22). To this end, the algorithm first selects the cluster C_{UL} that has the minimum *UL* when the diagnosis codes are anonymized (Step 19). The anonymization is performed by an adaptation of the CBA algorithm [45] (see Section 2.2). Our adaptation enforces k^m-anonymity to a cluster of records, instead of preventing re-identification based on the specified sets of diagnosis codes as CBA does (a detailed explanation is provided in Appendix A). Following, in Step 20, ARTUC searches for a cluster C'_{UL} which, if merged with C_{UL}, results in a cluster that has:

1. minimum *UL* (when generalized with our adaptation of the CBA algorithm), and
2. *NCP* at most δ (when generalized with Definition 2).

Intuitively, C'_{UL} is the "closest" cluster to C_{UL} with respect to *UL* that does not violate the *NCP* bound set by Requirement 1 of Problem 1. If C'_{UL} exists, the two clusters are merged and the algorithm attempts to merge another pair of clusters. Otherwise, the cluster merging stops.

(k, k^m)-anonymization. The (k, k^m)-anonymization phase is performed in Steps 24 to 27. In this phase, ARTUC generalizes the diagnosis codes in a cluster, using our adaptation of the CBA algorithm (Step 25), and checks whether the total number of suppressed diagnosis codes exceeds the parameter ϵ. In this case, the utility constraint set cannot be satisfied,

and the algorithm terminates (Step 26). Otherwise, the cluster is added into an initially empty set $\mathcal{D}$ that stores the output, and the next cluster is considered.

After all clusters have been considered, ART_{UC} returns the set $\mathcal{D}$, which is comprised of all clusters and is a (k, k^m)-anonymous version of the RT-dataset D (Step 27).

ART_{UC} is efficient and scales well with respect to the dataset size and anonymization parameters. A time complexity analysis of our algorithm can be found in Appendix C.

4.2. Example using ART_{UC}

To illustrate the operation of the ART_{UC} algorithm, we apply it to enforce $(2, 2^2)$-anonymity on the dataset of Figure 1a using the utility constraint set of Table 4. We set the parameters of ART_{UC} as follows: $k = 2$, $m = 2$, $\delta = 0.6$, and $\epsilon = 2$.

During the record grouping phase, the records of the dataset are split into the groups $G_1 = \{0,1,2,3,6,7\}$ and $G_2 = \{4,5\}$ (Step 1). G_1 is formed around the utility constraint u_1 (or equivalently around u_2 or u_3, which have the same elements in {Age,Origin,Gender} with u_1), while G_2 is formed around u_4. Next, ART_{UC} considers G_1 and creates the first cluster, C_1 (Steps 4 to 10). Let us assume that the record 0 is selected and moved to C_1, in Step 6. Then, the record 1 is added into C_1, because generalizing the demographic values of the records 0 and 1 together results in the minimum NCP value (Steps 7–9). Similarly, ART_{UC} creates the clusters C_2 and C_3, which contain the records $\{2,3\}$ and $\{6,7\}$ of G_1, respectively.

After that, the algorithm considers the second group, G_2, and creates the cluster $C_4 = \{4,5\}$. Since all records of the groups are assigned to clusters, Steps 11–14 are skipped. Then, ART_{UC} generalizes the demographic attributes of each cluster using Definition 2 (Step 15), as shown in Figure 1a, and proceeds into the next phase, because the NCP of the set of clusters $\mathcal{C} = \{C_1, C_2, C_3, C_4\}$ is lower than δ (Step 16).

In the cluster merging phase, the cluster C_1 has the minimum UL when generalized (Step 19). Then, the algorithm finds the cluster C_2, which satisfies both conditions (i.e., (a) $C_1 \cup C_2$ has minimum UL, and (b) NCP of $\mathcal{C} - C_1 - C_2 \cup \{C_1 \cup C_2\}$ is at most δ) (Step 20) and merges C_1 and C_2 (Step 21). Subsequently, ART_{UC} proceeds into the following phase, because any further merging of clusters results in a set of clusters having a higher NCP than δ (i.e., violates the Requirement 1 of Problem 1).

In the (k, k^m)-anonymization phase (Steps 24–26), the diagnosis codes in each cluster are generalized, and the (k, k^m)-anonymous RT-dataset, shown in Figure 3, is returned (Step 27).

4.3. Adaptation of ART_{UC} *based on progressive relaxation of Requirements 1 and 4*

As can be seen in the pseudocode, our algorithm produces an anonymized dataset that satisfies Requirements 1, 3, and 4 of Problem 1 and has minimal UL, except in the following cases:

- the NCP of the anonymized dataset with respect to the demographics exceeds the specified δ (i.e., Requirement 1 is not satisfied), or
- the number of suppressed diagnosis codes exceeds the specified value of ϵ (Requirement 4 is not satisfied).

In these cases, the satisfaction of the requirements is treated as a hard constraint, and the algorithm does not produce an anonymized dataset. This algorithmic design choice clearly reflects the semantics of utility constraints, which dictate that a group of records that is "too" generalized is useless for analysis, and it is consistent with prior work [46, 47, 48].

An alternative design choice is to relax Requirements 1 and 4 of Problem 1 progressively, as the algorithm is being executed, and let the data owner decide when the produced anonymized dataset is useful for them (i.e., it has acceptable NCP a sufficiently low number of suppressed diagnosis codes). This adaptation of ART_{UC} avoids the need for specifying δ and ϵ a priori and turns ART_{UC} into an "any time" method, in the spirit of the method of [20]. The adaptation is straightforward, and it is performed by replacing the Steps 16 to 25 of ART_{UC} with the pseudocode provided in Algorithm 2 below. In summary, the adaptation performs cluster merging and (k, k^m) anonymization iteratively. The merging of two clusters increases the NCP of the anonymized dataset, and it is performed as long as the NCP and number of suppressed diagnosis codes are deemed acceptable by the data owner. In the following, we discuss the adaptation in more detail.

As can be seen in the pseudocode of Algorithm 2, an anonymized dataset $\mathcal{D}$, comprised of all the formed clusters, is created (steps 1 to 2), and the data owner is asked whether the NCP of $\mathcal{D}$ is sufficiently low (step 3). If not, $false$ is returned (step 4), since subsequent cluster merging cannot decrease NCP, as mentioned above. Otherwise, the adaptation proceeds into the cluster merging phase (steps 5 to 21). In this phase, the clusters with the same generalized values in all demographic attributes are combined (step 6), as in ART_{UC}. Then, the clusters are merged iteratively (steps 7 to 21). Differently from ART_{UC}, however, each cluster, including the temporarily merged cluster C'_{UL}, is (k, k^m)-anonymized (steps 12 to 13), and the data owner is asked whether the NCP of the resultant dataset is sufficiently low (step 14). If it is, the clusters and dataset are updated (steps 15 to 16), and we proceed

```
1  Anonymize each cluster in 𝒞 using our adaptation of the CBA algorithm
2  𝒟 ← the set of anonymized clusters in 𝒞
3  if NCP(𝒞) is not deemed acceptable by the data owner then
4      return false
5  else
6      Merge all clusters of 𝒞 with the same generalized values in all the demographic attributes
7      repeat
8          Select the cluster C_UL ∈ 𝒞 with minimum UL when generalized with our adaptation of the CBA algorith .
9          Search for a cluster C'_UL ∈ 𝒞 − C_UL such that C_UL ∪ C'_UL has minimum UL (when generalized with our
             adaptation of the CBA algorithm)
10         if C'_UL exists then
11             temp ← 𝒞 − C_UL − C'_UL ∪ {C_UL ∪ C'_UL}
12             Apply the generalization of Definition 2 to each cluster in temp
13             Anonymize each cluster in temp using our adaptation of the CBA algorithm
14             if NCP(temp) is deemed acceptable by the data owner then
15                 Remove from 𝒞 clusters C_UL and C'_UL and add cluster C_UL ∪ C'_UL
16                 Update 𝒟
17             else if the number of suppressed diagnosis codes is acceptable then
18                 return 𝒟
19             else
20                 return false
21     until the set of clusters 𝒞 does not change;
22 return 𝒟
```

Algorithm 2: Adaptation of ART$_{UC}$ that progressively relaxes Requirements 1 and 4 of Problem 1.

into the next iteration. Otherwise, the *NCP* is deemed unacceptable, and the data owner is asked whether the number of suppressed diagnosis codes is sufficiently low (step 17). If it is, the clusters are not merged and an anonymized dataset with acceptable *NCP* and number of suppressed diagnosis codes is returned (step 18). Otherwise, *false* is returned, since further cluster merging and (k, k^m)-anonymization can only increase the number of suppressed diagnosis codes (steps 19 to 20). The cluster merging continues as long as the *NCP* is acceptable and the set of created clusters does not change (step 21). After that, the anonymized dataset, which has acceptable *NCP* and number of suppressed diagnosis codes, is returned (step 22).

5. Experimental evaluation

In this section, we evaluate ART$_{UC}$, in terms of data utility preservation and efficiency. We compare ART$_{UC}$ with two anonymization algorithms that are applicable to *RT*-datasets, namely **R**M$_R$ and BASELINE. **R**M$_R$ [61] aims to minimize the overall information loss without considering the intended analysis requirements (see Section 2.1). BASELINE is a baseline

Dataset	$\|D\|$	# of demographics	# of distinct diagnosis codes	Max, Avg # diagnosis codes per record
EHRD	208,387	2	13,963	185, 16.21
INFORMS	36,553	5	619	17, 4.27

Table 5: Description of EHRD and INFORMS dataset.

method, which performs record grouping and (k, k^m)-anonymization. BASELINE is similar in principle to the algorithm of Takahashi et al. [68], in that it pre-generalizes the demographic attributes. The BASELINE algorithm works as follows. First, it creates a set of groups of records, each of which corresponds to a different utility constraint. Then, it considers each group and generalizes the demographic attributes and diagnosis codes of every record in the group. The demographic attributes are generalized as specified by the utility constraint corresponding to the group (i.e., the value in each attribute is replaced by the corresponding value of the utility constraint of the group), while the diagnosis codes are anonymized following the Steps 24–26 of Algorithm ART$_{UC}$. BASELINE outperforms the algorithm of [68] in terms of preserving data utility, because it employs (k, k^m)-anonymity instead of k-anonymity and set-based generalization instead of hierarchy-based generalization for diagnosis codes.

5.1. Experimental setup

We have implemented all algorithms in C++ and ran all experiments on an Intel i7 at 3.2 GHz with 32 GB of RAM running Mac OS X 10.8. In our experiments, we use two *RT*-datasets, namely EHRD and Informs. Each record in these datasets contains demographics and ICD-9 codes.The EHRD dataset is proprietary (provided by a university medical center). The Informs dataset is publicly available at https://sites.google.com/site/informsdataminingcontest/data. The processing and analysis of these datasets was performed in a privacy-preserving way, according to standard practicies and policies. Table 5 summarizes the characteristics of the EHRD and INFORMS datasets.

In our experiments, the default anonymization values were set as follows:

$$k = 10, \quad m = 2, \quad \delta = 0.04, \quad and \quad \epsilon = 0.06 \cdot |\#distinctcodes|$$

The hierarchies were constructed as in [72]. To experiment with different types of utility requirements, we use 9 different utility constraint sets (illustrated in Table 6). The default utility constraint set for EHRD is UC_1 (see Table 6a). For instance, each utility constraint in this utility constraint set is comprised of:

UC_1	u_1={[0:12),All,Chapter$_1$}, u_2={[12:24),All,Chapter$_2$},...
UC_2	u_1={[0:12),All,Section$_1$},u_2={[12:24),All,Section$_2$},...
UC_3	u_1={[0:12),All,{001.*}},u_2={[12:24),All,{002.*}},...
UC_4	u_1={[0:24),All,Chapter$_1$}, u_2={[24:48),All,Chapter$_2$ ...
UC_5	u_1={[0:24),All,Section$_1$},u_2={[24:48),All,Section$_2$},...
UC_6	u_1={[0:24),All,{001.*}},u_2={[24:48),All,{002.*}},...
UC_7	u_1={[0:50),All,Chapter$_1$}, u_2={[50:150),All,Chapter$_2$},...
UC_8	u_1={[0:50),All,Section$_1$},u_2={[50:150),All,Section$_2$},...
UC_9	u_1={[0:50),All,{001.*}},u_2={[50:150),All,{002.*}},...

(a)

UC_1	u_1={*All*, [1918 : 1938),[0 : 18), [0 : 10), *All*, *Chapter*$_1$}, u_2={*All*, [1918 : 1938),[18 : 32), [0 : 10), *All*, *Chapter*$_1$},...
UC_2	u_1={*All*, [1918 : 1938),[0 : 18), [0 : 10), *All*, *Sections*$_1$}, u_2={*All*, [1918 : 1938), [18 : 32), [0 : 10), *All*, *Sections*$_1$},...
UC_3	u_1={*All*, [1918 : 1938),[0 : 18), [0 : 10), *All*, {001.∗}}, u_2={*All*, [1918 : 1938),[18 : 32), [0 : 10), *All*, {002.∗}},...
UC_4	u_1={*All*, [1918 : 1968),[0 : 18), [0 : 10), *All*, *Chapter*$_1$}, u_2={*All*, [1918 : 1968),[18 : 32), [0 : 10), *All*, *Chapter*$_1$},...
UC_5	u_1={*All*, [1918 : 1968),[0 : 18), [0 : 10), *All*, *Sections*$_1$}, u_2={*All*, [1918 : 1968),[18 : 32), [0 : 10), *All*, *Sections*$_1$},...
UC_6	u_1={*All*, [1918 : 1968),[0 : 18), [0 : 10), *All*, {001.∗}}, u_2={*All*, [1918 : 1968),[18 : 32), [0 : 10), *All*, {002.∗}},...
UC_7	u_1={*All*, [1918 : 2001),[0 : 18), [0 : 10), *All*, *Chapter*$_1$}, u_2={*All*, [1918 : 2001),[18 : 32), [0 : 10), *All*, *Chapter*$_1$},...
UC_8	u_1={*All*, [1918 : 2001),[0 : 18), [0 : 10), *All*, *Sections*$_1$}, u_2={*All*, [1918 : 2001),[18 : 32), [0 : 10), *All*, *Sections*$_1$},...
UC_9	u_1={*All*, [1918 : 2001),[0 : 18), [0 : 10), *All*, {00.∗}}, u_2={*All*, [1918 : 2001),[18 : 32), [0 : 10), *All*, {002.∗}},...

(b)

Table 6: Utility constraint sets used in our experiments with (a) the EHRD dataset, and (b) the INFORMS dataset. In each utility constraint set, the interval [x:y) contains all integers that are at least x and smaller than y. Chapter$_1$ (respectively, Section$_1$ and {001.*}) denotes the set of diagnosis codes whose immediate ancestor is the first Chapter (respectively, Section and 3-digit code 001) in the ICD-9 taxonomy [1].

- an interval [0:12), containing 12 consecutive values in Age, which are at least 0 and less than 12,
- the value All in Gender, and
- all diagnosis codes belonging in the same Chapter in the ICD-9 taxonomy [1].

The ICD-9 taxonomy organizes diagnosis codes into more general, semantically related concepts. The ICD-9 taxonomy for the diagnosis codes that are contained in the dataset of Figure 1a is depicted in Figure 6. The leaves contain the most detailed diagnosis codes, which have 4 or 5 numerical digits, and the immediate ancestors of leaves are 3-digit diagnosis codes. For example, the diagnosis code 053.20 (Herpes zoster dermatitis of eyelid) is a leaf, whose immediate ancestor is the 3-digit diagnosis code 053 (Herpes zoster). In turn, the immediate ancestors of 3-digit codes are Sections and those of Sections are Chapters. For example, the immediate ancestor of 053 is the Section {050, . . . , 059} (Viral diseases accompanied by exanthem), and the immediate ancestor of the latter is the Chapter {001,. . . ,139} (Infectious and parasitic diseases). Last, the immediate ancestor of all Chapters is the root value of the taxonomy, which represents any ICD-9 diagnosis code.

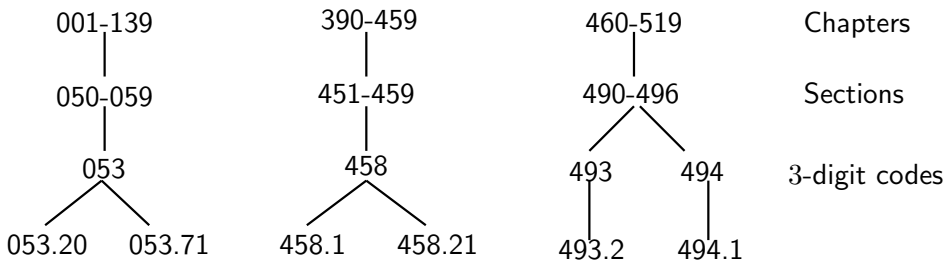


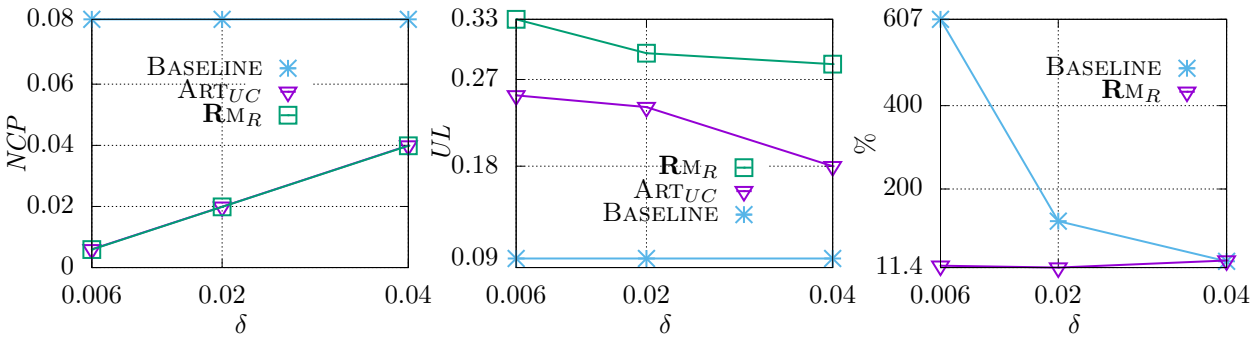
Figure 6: ICD-9 taxonomy of diagnosis codes in the dataset of Figure 1a.

Note that the utility constraint sets UC_1 and UC_3 in Table 6a contain the same elements in the demographic attributes but different elements in the set-valued attribute. Furthermore, $\mathsf{Chapter}_1$ of the utility constraint u_1 in UC_1 contains all the diagnosis codes in {001.*} of u_1 in UC_3, which represents the codes with immediate ancestor 001. Thus, UC_1 is less restrictive than UC_3 and is expected to be satisfied with less generalization. For the same reason UC_1 is less restrictive than UC_2, which is less restrictive than UC_3. Similar relations exist among UC_4 to UC_6 and among UC_7 to UC_9 in Table 6a. In addition, it is easy to see that u_1 in UC_7 is less restrictive than u_1 in UC_1, because its element [0:50) in Age contains the element [0:12) of the utility constraint u_1 in UC_1.

To evaluate data utility, we use the *NCP*, *UL*, and *ARE* measures (see Section 3.2). Lower values of these measures indicate lower data distortion and are preferable. Note that none of the methods optimizes *ARE* directly, since the methods do not take as input the query workloads. We use workloads of 100 queries, involving demographics and/or diagnosis codes, which retrieve random values and/or sets of 2 diagnosis codes by default, following [39, 47]. Since *ARE* reflects the average number of records that are retrieved incorrectly as part of answering a workload of queries, low *ARE* scores imply that anonymized data can be used to estimate the number of patients having certain demographic values and diagnosis codes fairly accurately.

5.2. Data utility comparison

We first evaluate all methods, with respect to the *NCP* and the *UL* measure, for varying δ. The results are reported in Figures 7 - 8. Increasing δ leads both ART_{UC} and $\mathbf{R}\text{M}_R$ to create larger clusters, which have higher *NCP* and lower *UL*. On the other hand, BASELINE is not affected by δ, because it creates clusters of fixed size, which favor the anonymization of diagnosis codes with low *UL*. In more detail, ART_{UC} outperformed BASELINE in *NCP*



(a) *NCP* comparison **(b)** *UL* comparison **(c)** Additional data utility loss

Figure 7: Utility comparison of BASELINE, $\mathbf{R}\text{M}_R$ and ART_{UC} for the EHRD dataset.

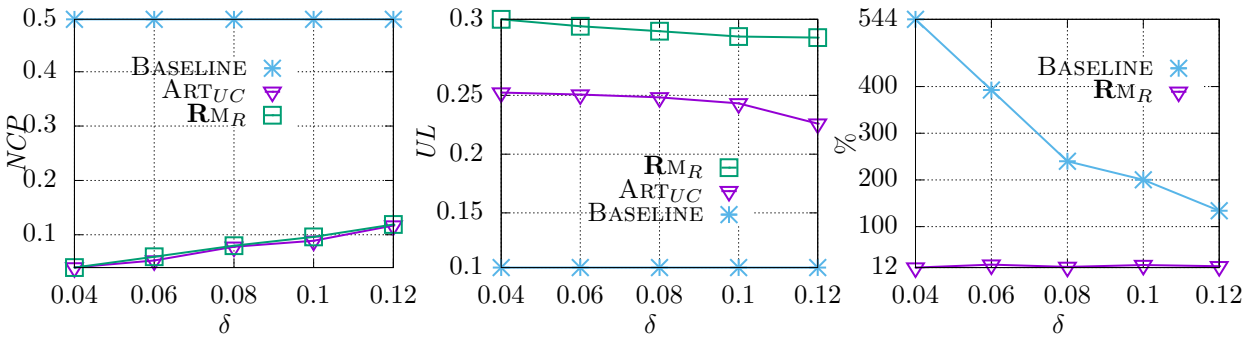
when we used EHRD dataset, achieving lower scores by 620% on average, while its *UL* scores were higher by 240% on average. The results for the INFORMS dataset, shown in Figure 8, are qualitatively similar to those of Figure 7. Specifically, ART_{UC} achieved lower scores by 662.07% on average, while its *UL* scores were higher by 138.01% on average. This indicates that the pre-generalization of demographic attributes, used by BASELINE, incurs high information loss, unlike the cluster formation and cluster merging strategies that are employed by ART_{UC}. In addition, the *NCP* scores of ART_{UC} and $\mathbf{R}\text{M}_R$ were equal and the *UL* scores of ART_{UC} were 50% better on average (and up to 92% better, for the case of EHRD, and 32.24% better on average for the case of INFORMS). This shows that ART_{UC} is able to satisfy the utility constraint set, while preserving the information in both demographics and diagnosis codes. Moreover, Figure 7c presents the *additional data utility loss* incurred by using $\mathbf{R}\text{M}_R$ or BASELINE instead of ART_{UC}. The additional data utility loss for $\mathbf{R}\text{M}_R$ is computed as:

$$\frac{1}{2} \cdot \left(\frac{NCP(\mathcal{D}_{\mathbf{R}\text{M}_R}) - NCP(\mathcal{D}_{\text{ART}_{UC}})}{NCP(\mathcal{D}_{\text{ART}_{UC}})} + \frac{UL(\mathcal{D}_{\mathbf{R}\text{M}_R}) - UL(\mathcal{D}_{\text{ART}_{UC}})}{UL(\mathcal{D}_{\text{ART}_{UC}})}\right) \cdot 100\%$$

The computation assumes that *NCP* and *UL* are equally important and is similar for BASELINE. As can be seen in Figure 7c, the additional data utility loss for $\mathbf{R}\text{M}_R$ is 18.6% on average and that for BASELINE is 252.1% on average. The result in Figure 8c is similar.

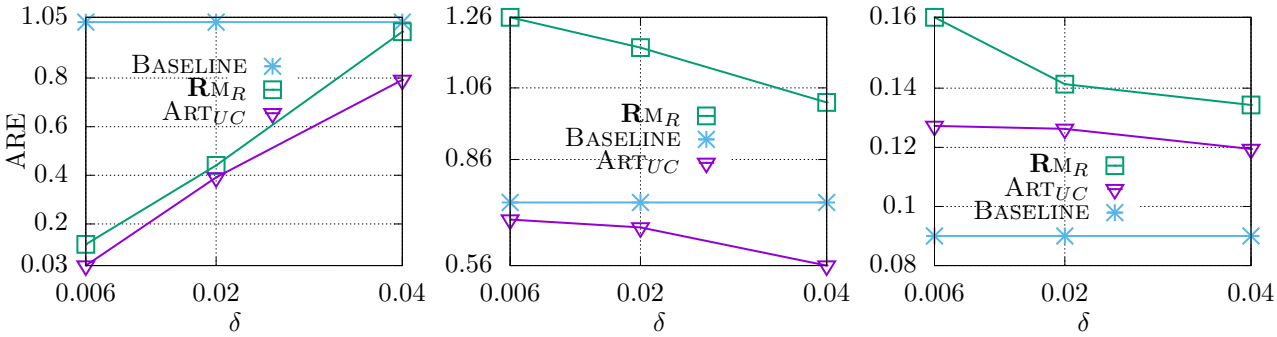
In the following experiments, we evaluate all methods, with respect to the *ARE* measure. The results are shown in Figures 9 - 10. For each dataset, we use three query workloads, namely $\mathcal{W}_1$, $\mathcal{W}_2$, and $\mathcal{W}_3$. In summary:

- The queries in $\mathcal{W}_1$ retrieve information based on the values of two demographics attributes.



(a) *NCP* comparison **(b)** *UL* comparison **(c)** Additional data utility loss

Figure 8: Utility comparison of BASELINE, $\mathbf{R}\text{M}_R$ and ART_{UC} for the INFORMS dataset.



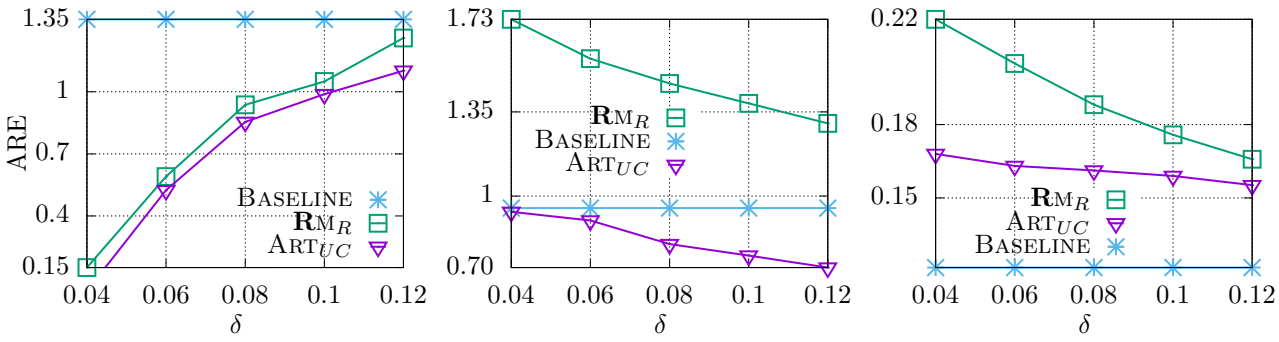
(a) Workload $\mathcal{W}_1$ **(b)** Workload $\mathcal{W}_2$ **(c)** Workload $\mathcal{W}_3$

Figure 9: Utility comparison using *ARE* for the EHRD dataset.

- The queries in $\mathcal{W}_2$ retrieve information based on the value of one demographic attribute and one diagnosis code.
- The queries in $\mathcal{W}_3$ retrieve information based on two diagnosis codes.

For example, a query in $\mathcal{W}_2$, which was used in the experiments with the EHRD dataset, can retrieve the number of patients associated with Age=32 and Chronic obstructive asthma.

As expected, the effectiveness of the evaluated methods is affected by the query workloads. For example, BASELINE, which favors the anonymization of diagnosis codes, has the worst effectiveness for $\mathcal{W}_1$ and the best for $\mathcal{W}_3$ (see Figures 9a and 9c). ART_{UC} is more effective than BASELINE for both $\mathcal{W}_1$ (e.g., 812% better, on average in the case of EHRD and 736.60% better on average, in the case of INFORMS) and $\mathcal{W}_2$ (34% better, on average, in the case of EHRD, and 17.53% better, on average, in the case of INFORMS). These results highlight that our method allows accurately answering queries that involve demographics



(a) Workload $\mathcal{W}_1$ **(b)** Workload $\mathcal{W}_2$ **(c)** Workload $\mathcal{W}_3$

Figure 10: Utility comparison using *ARE* for the INFORMS dataset.

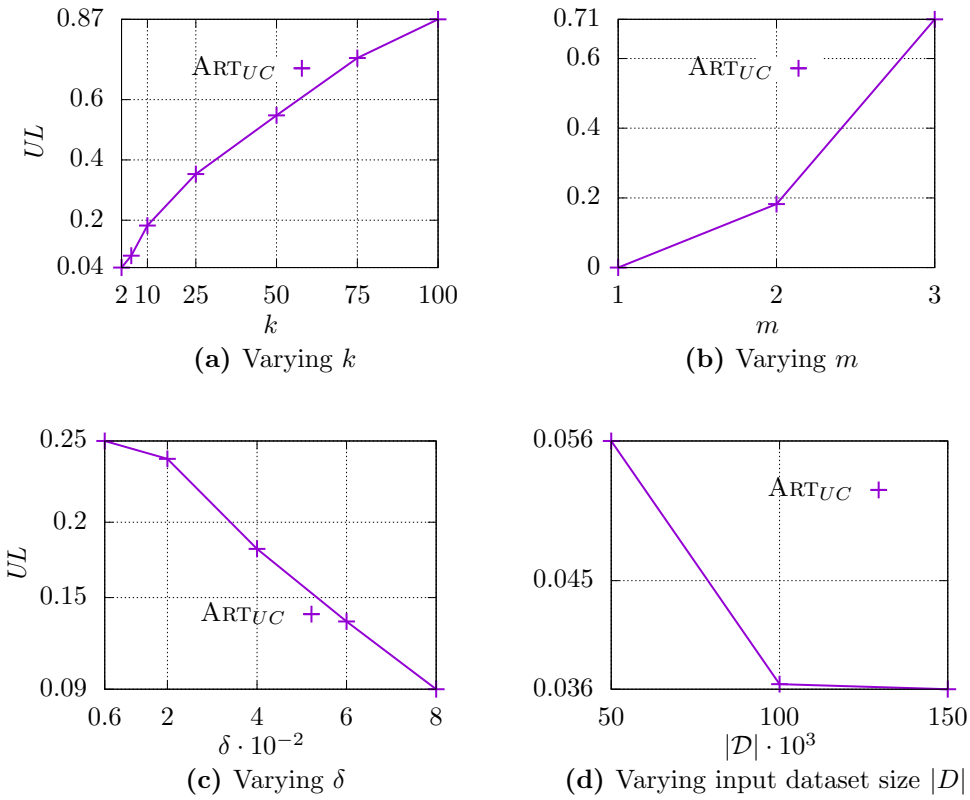
and diagnosis codes, which are the most important when publishing *RT*-datasets. On the other hand, $\mathbf{R}\text{M}_R$ merges together clusters that have semantically distant values in the demographic attributes, and this results in high *ARE* scores, particularly for $\mathcal{W}_2$ and $\mathcal{W}_3$ (see Figures 9b and 9c for EHRD and Figures 10b and 10c for INFORMS). For example, note in Figures 9b and 9c, that ART_{UC} is more effective than $\mathbf{R}\text{M}_R$ for $\mathcal{W}_1$ (47% better, on average), $\mathcal{W}_2$ (123% better, on average), and $\mathcal{W}_3$ (14% better on average).

In summary, the results in Figures 7, 8, 9 and 10 demonstrate that ART_{UC} preserves data utility better than BASELINE and $\mathbf{R}\text{M}_R$. In particular, BASELINE did not permit accurate analysis on demographics, while $\mathbf{R}\text{M}_R$ did not support the intended analysis tasks. Having established that ART_{UC} is more accurate than BASELINE and $\mathbf{R}\text{M}_R$, we will only report results for ART_{UC} in the following sections. In addition, the results using the EHRD dataset are quantitatively similar to those using the Informs dataset. Thus, for brevity, we will only report results for EHRD in the following sections.

5.3. Data utility evaluation of ART_{UC} *(fixed utility constraint set, varying parameters)*

In this section, we examine the data utility offered by our method, when there is a fixed utility constraint set and varying anonymization parameters. Specifically, we use the utility constraint set UC_1 of Table 6 and report results for *UL* and *ARE*, when we vary the size of the input dataset $|D|$ and the values of parameters k, m and δ parameters.

In the first set of experiments, we evaluate the information loss on diagnosis codes, using the *UL* measure. Specifically, we varied the parameters k, m, δ, as well as the size of the input *RT*-dataset $|D|$, and present the results with respect to *UL* in Figure 11. As can be seen in Figures 11a and 11b, *UL* values increase with k and m. Larger values of k and m



(a) Varying k **(b)** Varying m

(c) Varying δ **(d)** Varying input dataset size $|D|$

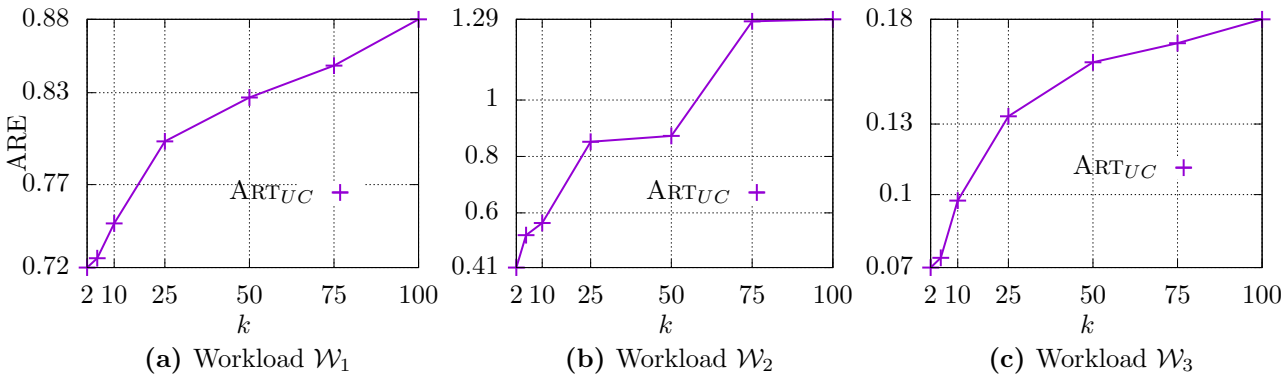
Figure 11: Utility comparison using UL

enforce stricter privacy requirements. To satisfy these requirements, ART_{UC} applied more generalization which led to higher formation loss.

On the contrary, as illustrated in Figure 11c, the UL scores are lower, for larger δ values. This is because a larger δ leads our method to create larger clusters, which favor the anonymization of diagnosis codes with low information loss. Similarly, increasing the dataset size, resulted in lower UL scores, as shown in Figure 11d. This is because larger datasets contain combinations of diagnosis codes which appear in more records[3]. Thus, the anonymization of these datasets can be performed with lower information loss.

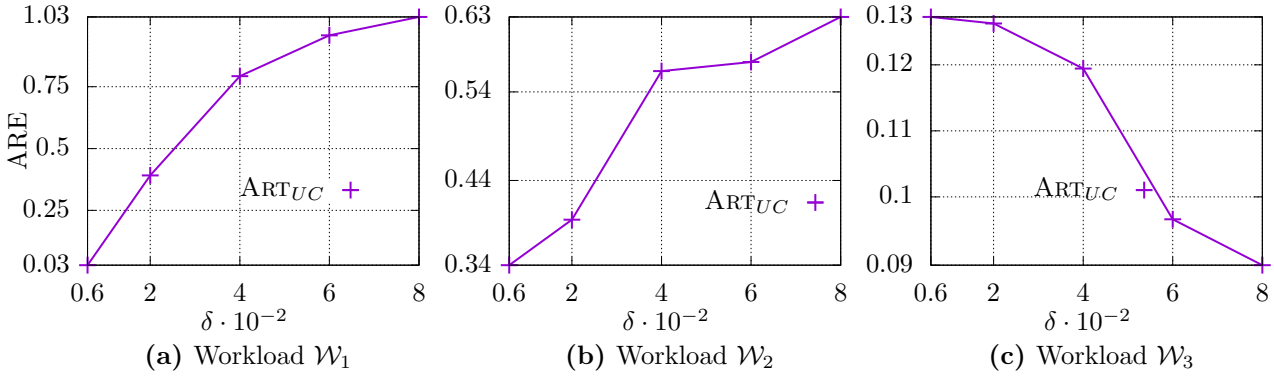
In the following experiments, we evaluate data utility, using the ARE measure, and the workloads $\mathcal{W}_1$, $\mathcal{W}_2$, and $\mathcal{W}_3$, used in Section 5.2. Specifically, Figures 12a, 12b, and 12c show the ARE scores, for varying k, and for the workloads $\mathcal{W}_1$, $\mathcal{W}_2$, and $\mathcal{W}_3$, respectively. Observe that larger values for k, as expected, increase information loss. However, the ARE scores

[3]In this experiment, each dataset contains randomly selected records, and larger datasets contain all records of the smaller datasets.



(a) Workload $\mathcal{W}_1$ **(b)** Workload $\mathcal{W}_2$ **(c)** Workload $\mathcal{W}_3$

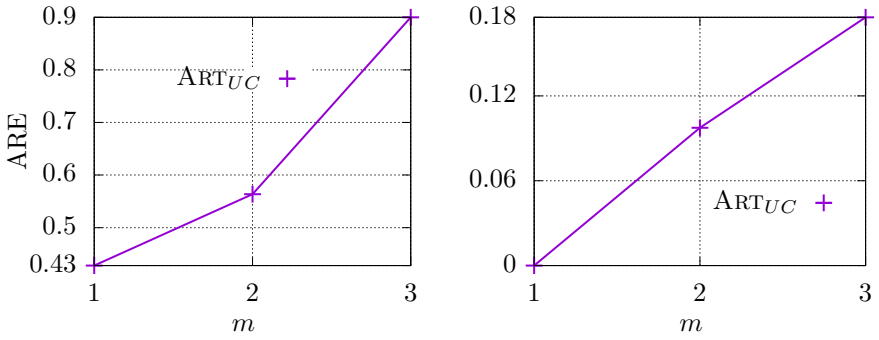
Figure 12: *ARE* measure for varying k.



(a) Workload $\mathcal{W}_1$ **(b)** Workload $\mathcal{W}_2$ **(c)** Workload $\mathcal{W}_3$

Figure 13: *ARE* measure for varying δ.

remain relatively low, even when $k = 100$, which is 20 times larger than the commonly used value for k [46]. Figures 13a, 13b, and 13c illustrate the *ARE* scores, for varying δ, and for the workloads $\mathcal{W}_1$, $\mathcal{W}_2$, and $\mathcal{W}_3$, respectively. The tested values of δ were in $[6 \cdot 10^{-3}, 8 \cdot 10^{-2}]$, which correspond to the minimum and maximum number of clusters constructed by our method, respectively. *NCP* did not exceed δ in all tested cases. As can be seen, increasing δ leads to higher *ARE* scores for $\mathcal{W}_1$ and $\mathcal{W}_2$ but to lower *ARE* scores for $\mathcal{W}_3$. This is because, ARTUC creates larger clusters of similar diagnosis codes when δ is larger, which lead to higher information loss on the demographic attributes but to lower information loss on the diagnosis codes. We also report the *ARE* scores for varying m in Figures 14a and 14b, which correspond to $\mathcal{W}_2$ and $\mathcal{W}_3$, respectively. Since varying m only affects the information loss on diagnosis codes, we do not report the result for $\mathcal{W}_1$. Note that increasing m results in larger *ARE* scores. This is because more combinations of diagnosis codes need protection



(a) Workload $\mathcal{W}_2$ **(b)** Workload $\mathcal{W}_3$

Figure 14: *ARE* measure for varying m.

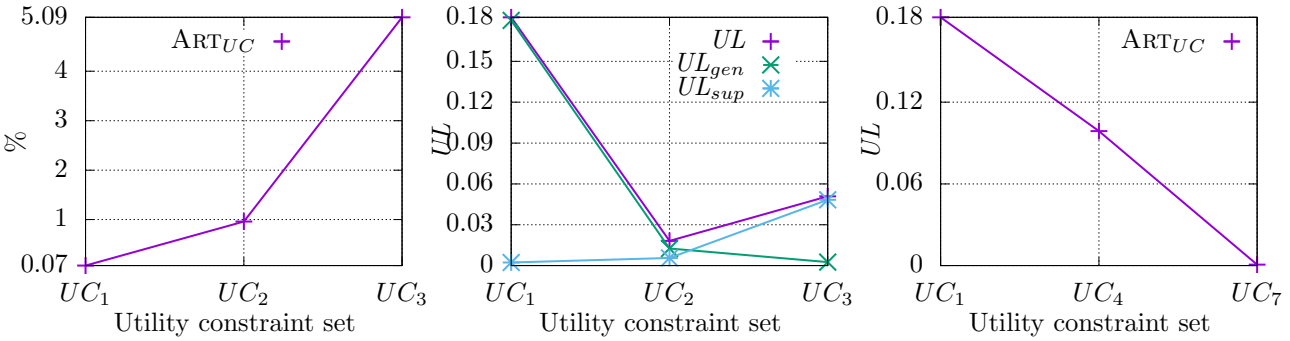
Setup	Utility constraint set UC
1	$UC_1 = \{$ {[0:12), All, Chapter$_1$}, {[12:24), All, Chapter$_2$}, ... $\}$
	$UC_2 = \{$ {[0:12), All, Sections$_1$}, {[12:24), All, Sections$_2$}, ... $\}$
	$UC_3 = \{$ {[0:12), All, {001.*}}, {[12:24), All, {001.*}}, ... $\}$
2	$UC_1 = \{$ {[0:12), All, Chapter$_1$}, {[12:24), All, Chapter$_2$}, ... $\}$
	$UC_4 = \{$ [0:24), All, Chapter$_1$}, {[24:48), All, Chapter$_2$, ... $\}$
	$UC_7 = \{$ {[0:50), All, Chapter$_1$}, {[50:150), All, Chapter$_2$}, ... $\}$

Table 7: Utility constraint sets, used in each setup (taken from Table 6).

when m is larger, and this leads ART$_{UC}$ to apply more generalization.

5.4. Data utility evaluation of ART$_{UC}$ *(varying utility constraint sets, fixed parameters)*

In this section, we examine the data utility offered by our method, when all parameters are fixed to their default values and various utility constraint sets are used. In particular, we study how the specified utility constraint set affects: (i) the percentage of suppressed diagnosis codes, (ii) the *UL* measure, and (iii) the *ARE* measure. We do not report results for *NCP*, because its value was equal to the specified threshold $\delta = 0.04$. We used two different setups of utility constraint sets, which are presented in Table 7. In Setup 1, ART$_{UC}$ is applied with UC_1, UC_2, or UC_3. In Setup 2, one of the utility constraint sets UC_1, UC_4, and UC_7 is used instead. As we move from left to right in the first row of Table 7, the utility constraint sets become more restrictive. That is, UC_1 is the least restrictive set among those in Setup 1 (see Section 5.1), UC_3 is the most restrictive, and UC_2 lies in between UC_1 and UC_3. On the contrary, as we move from left to right in the second row Table 7, the utility constraint sets become less restrictive. This is because they contain larger intervals in the demographic attribute Age.



(a) % of suppressed codes (Setup 1) **(b)** Measuring UL for Setup 1 **(c)** Measuring UL for Setup 2

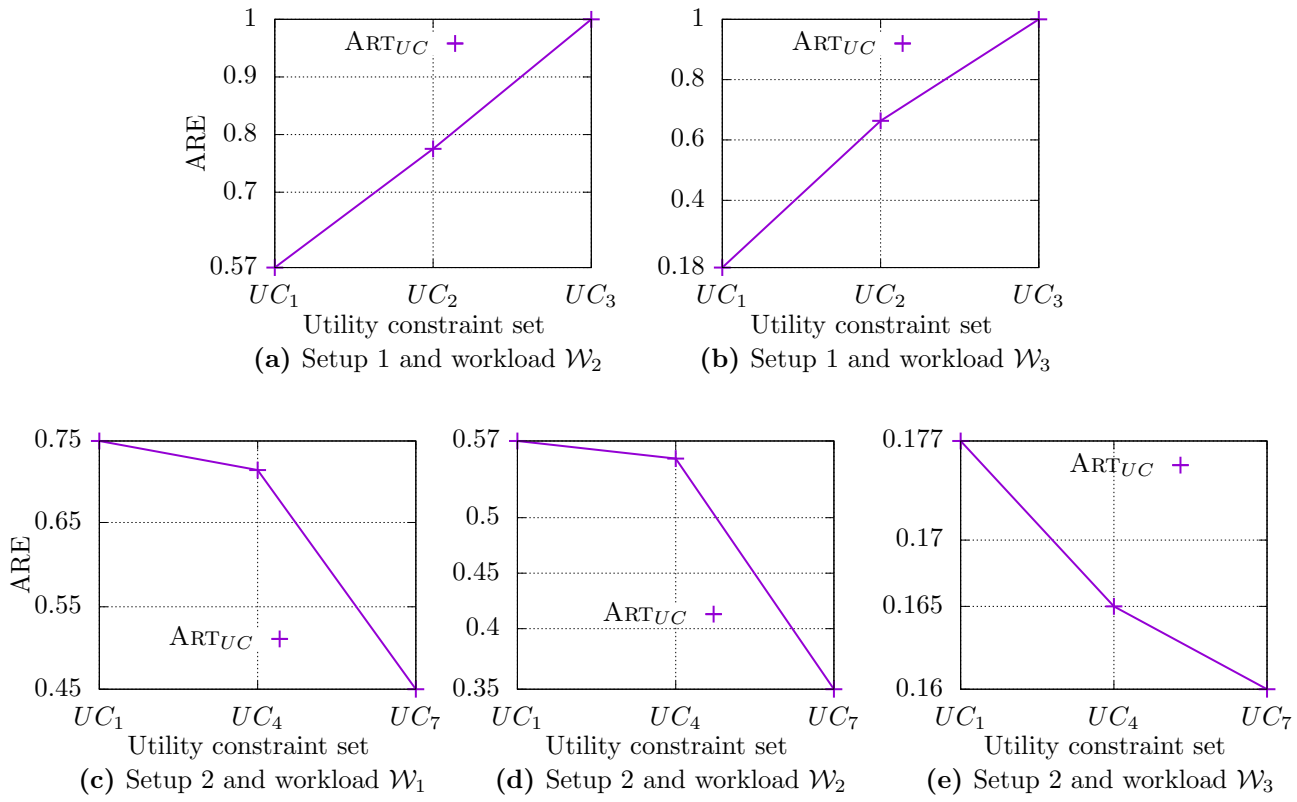
Figure 15: Data utility for different utility constraint sets (Setup 1 and 2).

In our first experiment, we report the percentage of suppressed diagnosis codes (Figure 15a). The percentage of suppressed diagnosis codes is the lowest for UC_1, the least restrictive utility constraint set in Setup 1, and it is the highest for UC_3. This is because restrictive utility constraint sets limit the number of allowable generalizations and force ART$_{UC}$ to apply suppression, in order to enforce (k, k^m) anonymity. Suppression is employed only when the use of generalization is not sufficient to (k, k^m)-anonymize a cluster. Thus, the percentage of suppressed diagnosis codes is relatively small (e.g., 1% for UC_2 and 5.09% for UC_3) and does not exceed the specified threshold ϵ.

In the following experiment, we consider the UL metric and in Figure 15b illustrate its values for the utility constraints sets of Setup 1. Note that UL values decrease between UC_1 and UC_2 and slightly increase between UC_2 and UC_3. To explain this behavior, we decompose UL into the utility loss caused by generalization, denoted by UL_{gen}, and the utility loss caused by suppression, denoted by UL_{sup} (obviously $UL = UL_{sup} + UL_{sup}$). As can be seen in Figure 15b, UL_{gen} decreases as the utility constrains become more restrictive (i.e., moving from UC_1 to UC_2 and UC_3), whereas UL_{sup} follows an opposite trend. This justifies that the increase in UL for UC_3 is a result of the use of suppression by ART$_{UC}$ (see also the percentage of suppression that corresponds to UC_3 in Figure 15a).

In Figure 15c, we use the utility constraint sets of Setup 2. These utility constraint sets were satisfied without suppression, thus, the UL score decreases as the utility constraint set becomes less restrictive (from the most strict UC_1 to the more loose UC_7).

Subsequently, we report the ARE scores for both setups. Figures 16a and 16b correspond to the utility constraint sets of Setup 1 and to the query workloads $\mathcal{W}_2$ and $\mathcal{W}_1$, respectively. ARE scores are the largest for UC_3, the most restrictive utility constraint set UC_3 of Setup



(a) Setup 1 and workload $\mathcal{W}_2$ **(b)** Setup 1 and workload $\mathcal{W}_3$

(c) Setup 2 and workload $\mathcal{W}_1$ **(d)** Setup 2 and workload $\mathcal{W}_2$ **(e)** Setup 2 and workload $\mathcal{W}_3$

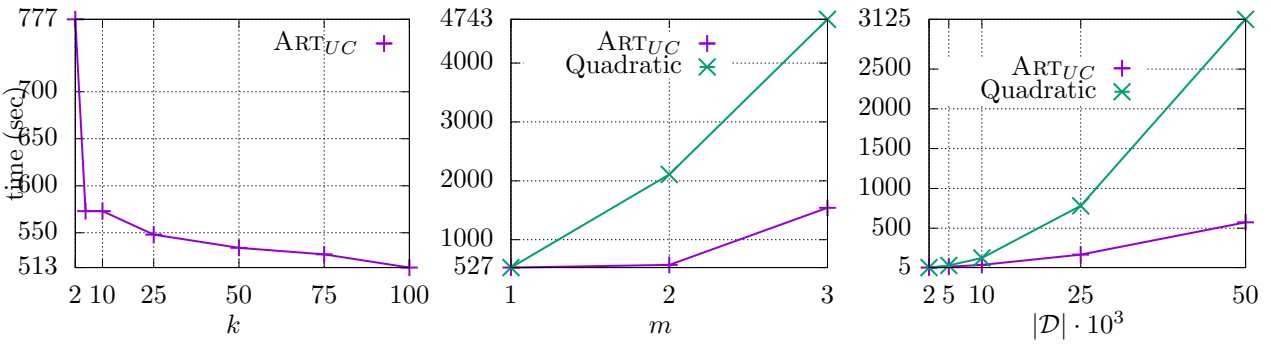
Figure 16: ARE for different utility constraint sets.

1 since to satisfy UC_3 more generalization and suppression is required. Similar results are in order for Setup 2; Figures 16c, 16d, and 16e correspond to the query workloads $\mathcal{W}_1$, $\mathcal{W}_2$, and $\mathcal{W}_3$, respectively. Specifically, ARE is larger for UC_1, which is the most restrictive utility constraint set of Setup 2.

In summary, the use of different utility constraint sets affects data utility. For instance, replacing the utility constraint UC_1 with UC_3, results in higher UL_{sup} and lower UL_{gen} scores. On the contrary, replacing UC_1 with UC_7, results in high NCP and low UL scores, as less restrictive constraints lead ART$_{UC}$ to create larger groups of records. However, in both cases, our method preserved utility well, according to all tested measures. Furthermore, our method was able to generate data that can be used to support the intended studies, as the utility constraints were always satisfied.

5.5. Efficiency evaluation of ART$_{UC}$

In this section, we evaluate the execution time of our method ART$_{UC}$, by varying the parameters k, m, and $|D|$ (size of dataset). The results are depicted in Figures 17a, 17b, and



(a) Varying k **(b)** Varying δ **(c)** Varying the dataset size $|D|$

Figure 17: Execution evaluation.

17c. Figure 17a shows that the runtime of our method decreases with k, as larger values of k result in larger initial clusters and fewer clusters merging operations. On the other hand, Figure 17b shows that runtime increases with m, as there are more combinations of diagnosis codes that must be protected. Notice though that the increase is subquadratic. Finally, Figure 17c shows that the execution time of our method increases with $|D|$[4]. The increase is also subquadratic in $|D|$. Overall, the results show that our method can anonymize datasets containing thousands of patient records within minutes.

6. Discussion

This section explains how our work can be extended to deal with different types of healthcare data and privacy requirements. In addition, it discusses limitations, which suggest opportunities for further research.

Our approach was developed for RT-datasets in which the set-valued attribute contains diagnosis codes. However, in certain applications, a patient record may also contain procedural codes (e.g., CPT codes which describe medical, surgical, and diagnostic services [55]), as well as medications. Extending our approach to deal with such datasets requires forming a set-valued attribute that contains all different types of codes. More formally, multiple set-valued attributes, $A_{l+1}, \ldots, A_{l+m}$, can be modeled as a single set-valued attribute A_L, whose domain (set of possible values) contains every value in the domain of A_{l+1} or A_{l+2} or $\ldots$ A_{l+m}, preceded by the domain name. For example, the domain of A_L contains a value $A_{l+1}.u$ to represent the value u of A_{l+1}. This is simply to distinguish between values

[4]In this experiment, we used random subsets of the dataset, whose records were contained in all larger sets.

of different attributes. In addition, each element v_{A_L} of a utility constraint must contain values of a single attribute, to prevent the generalization of values of different attributes (e.g., diagnosis and procedural codes), which are difficult to interpret.

Furthermore, our approach aims to prevent identity disclosure, which is the most important privacy requirement for healthcare data. This is suggested by the fact that "*all the publicly known examples of re-identification of personal information have involved identity disclosure*" [15] and by the fact that the majority of healthcare data anonymization methods focus on preventing identity disclosure. However, our algorithm can be extended to prevent attribute disclosure [72], when there are sensitive diagnosis codes with which patients are not willing to be associated. To achieve this, we may appropriately modify our adaptation of the CBA algorithm used in Steps 19, 20 and 25 of Algorithm ART$_{UC}$ (see also Appendix A) so that it enforces (k, l^m)-diversity [61] on the sensitive diagnosis codes of each cluster. The (k, l^m)-diversity principle guarantees that an attacker who knows all the values of the demographic attributes and up to m diagnosis codes of a patient cannot associate these diagnosis codes with any combination of sensitive diagnosis codes, with probability larger than $\frac{1}{l}$, where $l \geq 2$ is a parameter specified by data owners. This modification requires to apply additional generalization to ensure that any combination of p and sensitive diagnosis codes appears in at least l records of the anonymized cluster. Thus, this modification incurs additional information loss and computational overhead. The evaluation of the extended version of our algorithm that enforces (k, l^m)-anonymity is left as future work.

Moreover, alike other data anonymization methods, our approach assumes that data owners are able to select appropriate values for the parameters k and m, which model the privacy requirements, as well as for δ and ϵ, which model the maximum allowable level of information loss caused by generalizing demographic attributes and by suppressing diagnosis codes, respectively. However, configuring these parameters in an optimal manner, for a given RT-dataset, is not straightforward.

For example, the ART$_{UC}$ algorithm stops (i.e., the utility requirements are not satisfied) when: (I) the NCP of the anonymized dataset with respect to the demographics exceeds the specified δ (Step 16), or (II) the number of suppressed diagnosis codes exceeds the specified ϵ (Step 26). In these cases, the satisfaction of the utility requirements (requirements 1 and 3 of Problem 1) is treated as a hard constraint, and thus the algorithm does not produce an anonymized dataset. Our choice clearly reflects the semantics of utility constraints, which dictate that a group of records that is "too" generalized is useless for analysis, and is consistent with prior work [45, 46, 47].

An alternative choice would be to relax the utility requirements, by setting larger δ and ϵ. This raises the question of how to relax the requirements in a way that is easy and intuitive for the data owners, given that their specification is data-dependent. To address this question, we propose to involve the data owner in the execution of our algorithm, so that they can see select when the information loss with respect to the relational attributes and the number of suppressions are acceptable, as the algorithm progresses. This makes our algorithm an "any time" method, in the spirit of [20]. This can be easily done by the following two adaptations of our algorithm. The first adaptation is to perform generalization after the current step 3. The *NCP* of this dataset gives the maximum possible δ that the user can set. Then, to perform generalization of each cluster after its creation and output the current *NCP* of the resultant dataset, which will get smaller as more clusters as created. The data owner can stop the cluster formation phase, when the *NCP* is sufficiently small. After that, the algorithm will continue into cluster merging. The second adaptation is to output the total number of suppressed diagnosis codes after step 12 of **GenDiag**. The data owner can stop the execution of the algorithm when the number of suppressions is deemed "too" high. As shown in our experiments, the number of suppressions is very small or zero in practice, so the data owner will not have to examine the output many times, which makes the adaptation easy to use. We acknowledge that the data owner may need to examine other data quality indicators in addition to *NCP* and number of suppressed diagnosis codes. Towards this goal, we aim to incorporate our algorithm, together with the adaptations, in the SECRETA anonymization tool [60]. The tool offers a GUI that outputs many data utility indicators that may assist the data owner towards the specification of δ and ϵ.

Our approach aims to produce an anonymized *RT*-dataset that remains useful for intended analytic tasks, modeled with utility constraints, as well as for general analytic tasks. However, in a different setting, the anonymized dataset needs to remain useful for building a pre-determined data mining model (e.g., a classifier). In this setting, a different anonymization methodology which aims to preserve data utility for the specified data mining model (in the spirit of [53]) may preserve data utility better.

Last, our approach considers an unordered set of diagnosis codes, as the existing algorithms for anonymizing *RT*-datasets [31, 61] do. However, certain applications, such as longitudinal studies, require ordered sets (or ordered multisets) of diagnosis codes [69]. Anonymizing such data in a utility-preserving way is challenging because it requires preserving the sequentiality of data. To the best of our knowledge, the problem has not been considered and serves as an interesting avenue for future work.

7. Conclusions

Publishing datasets that contain demographics and diagnosis codes (RT-datasets) is important, in the context of several medical analysis tasks. To preserve the privacy and utility of RT-datasets, we proposed an approach that enforces (k, k^m)-anonymity, while satisfying intended analysis requirements with minimal information loss. In particular, we introduced the concept of utility constraints for RT-datasets, to limit the amount of data generalization, and developed an algorithm that constructs (k, k^m)-anonymous clusters of records, using generalization and suppression. Experiments using a dataset containing over $200,000$ electronic health records showed that our algorithm is effective at preserving data utility and also efficient.

References

[1] ICD-9 taxonomy of diseases. http://www.cdc.gov/nchs/data/icd/icd9cm_guidelines_2011.pdf.

[2] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference in Very Large Data Bases (VLDB)*, volume 1215, pages 487–499, 1994.

[3] Frank Boulton. Evidence-based criteria for the care and selection of blood donors, with some comments on the relationship to blood supply, and emphasis on the management of donation-induced iron depletion. *Transfusion Medicine*, 18(1):13–27, 2008.

[4] Ji-Won Byun, Ashish Kamra, Elisa Bertino, and Ninghui Li. Efficient k-anonymization using clustering techniques. In *Advances in Databases: Concepts, Systems and Applications*, pages 188–200, 2007.

[5] Jianneng Cao, Panagiotis Karras, Chedy Raïssi, and Kian-Lee Tan. *rho*-uncertainty: Inference-proof transaction anonymization. *Very Large Data Bases Endowment (PVLDB)*, 3(1):1033–1044, 2010.

[6] Donna Cartwright. Icd-9-cm to icd-10-cm codes: What? why? how? *Advances in wound care*, 2(10):588–592, 2013.

[7] Naveen Cotha, Parachur Kumar, and MarinaSokolova. Multi-label learning in classification of patients' quasi-identifiers. *Progress in Artificial Intelligence*, 4(3):37–48, 2015.

[8] Fida Kamal Dankar and Khaled El Emam. Practicing differential privacy in health care: A review. *Transactions on Data Privacy*, 6(1):35–67, 2013.

[9] Joshua C Denny. Chapter 13: mining electronic health records in the genomics era. *PLoS Computational Biology*, 8(12):e1002823, 2012.

[10] Josep Domingo-Ferrer and Josep MMateo-Sanz. Practical data-oriented microaggregation for statistical disclosure control. *Knowledge and Data Engineering, IEEE Transactions on*, 14(1):189–201, 2002.

[11] Kirsty Douglas, Laurann Yen, Rosemary Korda, Marjan Kljakovic, and Nicholas Glasgow. Chronic disease management items in general practice: a population-based study of variation in claims by claimant characteristics. *Medical Journal of Austalia*, pages 198–202, 2011.

[12] Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming*, pages 1–12, 2006.

[13] Anne Eder, Mindy Goldman, Susan Rossmann, Dan Waxman, and Celso Bianco. Selection criteria to protect the blood donor in north america and europe: past (dogma), present (evidence), and future (hemovigilance). *Transfusion Medicine Reviews*, 23(3):205–220, 2009.

[14] Gustaf Edgren, Henrik Hjalgrim, Marie Reilly, Trung Nam Tran, Klaus Rostgaard, Agneta Shanwell, Kjell Titlestad, Johanna Adami, Agneta Wikman, Casper Jersild, Gloria Gridley, Louise Wideroff, Olof Nyren, and Mads Melbye. Risk of cancer after blood transfusion from donors with subclinical cancer: a retrospective cohort study. *The Lancet*, 369(9574):1724–1730, 2007.

[15] Khaled El Emam. Methods for the de-identification of electronic health records for genomic research. *Genome Medicine*, 3(4):1–9, 2011.

[16] Khaled El Emam and Fida Kamal Dankar. Protecting privacy using k-anonymity. *Journal of the American Medical Informatics Association*, 15(5):627–637, 2008.

[17] IBM Institute for Business Value. The value of analytics in healthcare: From insights to outcomes, 2012, http://www-935.ibm.com/services/us/gbs/thoughtleadership/ibv-healthcare-analytics.html.

[18] National Center for Research Resources (US). Widening the use of Electronic Health Record data for research, 2009, http://videocast.nih.gov/summary.asp?live=8062.

[19] Jerome H. Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, September 1977.

[20] Benjamin Fung, Ke Wang, and Philip S Yu. Top-down specialization for information and privacy preservation. In *Proceedings of the 21st International Conference on Data Engineering (ICDE)*, pages 205–216. IEEE, 2005.

[21] Gabriel Ghinita, Panagiotis Karras, Panos Kalnis, and Nikos Mamoulis. Fast data anonymization with low information loss. In *33rd International Conference on Very Large Data Bases*, VLDB '07, pages 758–769, 2007.

[22] Gabriel Ghinita, Panagiotis Karras, Panos Kalnis, and Nikos Mamoulis. A framework for efficient data anonymization under privacy and accuracy constraints. *ACM Transactions on Database Systems (TODS)*, 34(2):9, 2009.

[23] Aristides Gionis, Amon Mazza, and Tamir Tassa. k-anonymization revisited. In *Proceedings of the 24th International Conference on Data Engineering (ICDE)*, pages 744–753. IEEE, 2008.

[24] Aris Gkoulalas-Divanis and Grigorios Loukides. Utility-guided clustering-based transaction data anonymization. *Transactions on Data Privacy*, 5(1):223–251, 2012.

[25] Aris Gkoulalas-Divanis, Grigorios Loukides, and Jimeng Sun. Publishing data from electronic health records while preserving privacy: A survey of algorithms. *Journal of Biomedical Informatics*, 50:4–19, 2014.

[26] David Goldsbury, Mark Fort Harris, Shane Pascoe, Ian Olver, Michael Barton, Allan Spigelman, and Dianne O'Connell. Socio-demographic and other patient characteristics associated with time between colonoscopy and surgery, and choice of treatment centre for colorectal cancer: a retrospective cohort study. *BMJ Open*, 2(3), 2012.

[27] Philippe Golle. Revisiting the uniqueness of simple demographics in the us population. In *2006 Workshop on Privacy in the Electronic Society*, pages 77–80. ACM Press, 2006.

[28] Medicare Plan Payment Group. Proposed Changes to the CMS-HCC Risk Adjustment Model for Payment, Year 2017, https://www.cms.gov/Medicare/Health-Plans/MedicareAdvtgSpecRateStats/Downloads/RiskAdj2017ProposedChanges.pdf.

[29] Jiawei Han and Micheline Kamber. *Data Mining, Southeast Asia Edition: Concepts and Techniques*. Morgan Kaufmann San Francisco, Calif, USA, 2006.

[30] Yeye He and Jeffrey F Naughton. Anonymization of set-valued data via top-down, local generalization. *Proceedings of the VLDB Endowment (PVLDB)*, 2(1):934–945, 2009.

[31] Raymond D. Heatherly, Grigorios Loukides, Joshua C. Denny, Jonathan L. Haines, Dan M. Roden, and Bradley A. Malin. Enabling genomic-phenomic association discovery without sacrificing anonymity. *PLoS ONE*, 8(2):e53875, 02 2013.

[32] INFORMS. Data Mining Contest, 2008, https://sites.google.com/site/informsdataminingcontest/.

[33] Tochukwu Iwuchukwu and Jeffrey F Naughton. k-anonymization as spatial indexing: Toward scalable and incremental anonymization. In *Proceedings of the VLDB Endowment (PVLDB)*, pages 746–757, 2007.

[34] Vijay S. Iyengar. Transforming data to satisfy privacy constraints. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 279–288, New York, NY, USA, 2002. ACM.

[35] Xiao Jia, Chao Pan, Xinhui Xu, Kenny Q Zhu, and Eric Lo. ρ-uncertainty anonymization by partial suppression. In *Database Systems for Advanced Applications*, pages 188–202. Springer, 2014.

[36] Anna Kemp, David Preen, Christobel Saunders, C. D'Arcy Holman, MaxBulsara, Kris Rogers, and Elizabeth Roughead. Ascertaining invasive breast cancer cases; the validity of administrative and self-reported data sources in australia. *BMC Medical Research Methodology*, 13(1):1–8, 2013.

[37] Michael Laszlo and Sumitra Mukherjee. Minimum spanning tree partitioning algorithm for microaggregation. *Knowledge and Data Engineering, IEEE Transactions on*, 17(7):902–911, 2005.

[38] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *ACM SIGMOD international conference on Management of data*, pages 49–60, 2005.

[39] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Mondrian multidimensional k-anonymity. In *Proceedings of the 22nd International Conference on Data Engineering (ICDE)*, pages 25–25. IEEE, 2006.

[40] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Workload-aware anonymization. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 277–286. ACM, 2006.

[41] Jiuyong Li, Raymond Wong, Ada Fu, and Jian Pei. Achieving k-anonymity by clustering in attribute hierarchical structures. *Data Warehousing and Knowledge Discovery*, pages 405–416, 2006.

[42] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. τ-closeness: Privacy beyond k-anonymity and ℓ-diversity. In *21st IEEE International Conference on Data Engineering (ICDE)*, volume 7, pages 106–115, 2007.

[43] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. Closeness: A new privacy measure for data publishing. *Knowledge and Data Engineering, IEEE Transactions on*, 22(7):943–956, 2010.

[44] Grigorios Loukides, Joshua C. Denny, and Bradley Malin. The disclosure of diagnosis codes can breach

research participants' privacy. *Journal of the American Medical Informatics Association*, 17(3):322–327, 2010.

[45] Grigorios Loukides and Aris Gkoulalas-Divanis. Utility-aware anonymization of diagnosis codes. *IEEE Journal of Biomedical and Health Informatics*, 17(1):60–70, Jan 2013.

[46] Grigorios Loukides, Aris Gkoulalas-Divanis, and Bradley Malin. Anonymization of electronic medical records for validating genome-wide association studies. *Proceedings of the National Academy of Sciences*, 17:7898–7903, 2010.

[47] Grigorios Loukides, Aris Gkoulalas-Divanis, and Bradley Malin. COAT: Constraint-based anonymization of transactions. *Knowledge and Information Systems*, 28(2):251–282, 2011.

[48] Grigorios Loukides, Aris Gkoulalas-Divanis, and Jianhua Shao. Anonymizing transaction data to eliminate sensitive inferences. In *Database and Expert Systems Applications*, pages 400–415. Springer, 2010.

[49] Grigorios Loukides, Aris Gkoulalas-Divanis, and Jianhua Shao. Efficient and flexible anonymization of transaction data. *Knowledge and information systems*, 36(1):153–210, 2013.

[50] Grigorios Loukides, John Liagouris, Aris Gkoulalas-Divanis, and Manolis Terrovitis. Disassociation for electronic health record privacy. *Journal of Biomedical Informatics*, 50:46–61, 2014.

[51] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[52] Stephane M. Meystre, Jeffrey Friedlin, Brett R. South, Shuying Shen, and Matthew H Samore. Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC Medical Research Methodology*, pages 10–70, 2010.

[53] Noman Mohammed, Benjamin Fung, Patrick CK Hung, and Cheuk kwong Lee. Anonymizing healthcare data: a case study on the blood transfusion service. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1285–1294. ACM, 2009.

[54] Noman Mohammed, Xiaoqian Jiang, Rui Chen, Benjamin C. M. Fung, and Lucila Ohno-Machado. Privacy-preserving heterogeneous health data sharing. *Journal of the American Medical Informatics Association*, 20(3):462–469, 2013.

[55] US National Library of Medicine. CPT codes. https://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/CPT/.

[56] Information Commissioners Office. U.K. Anonymization Code. Federal Register, vol. 67, no. 157, 2002, http://www.ico.org.uk/for_organisations/data_protection/topic_guides/anonymisation/.

[57] National Committee on Vital and Health Statistics. HIPAA code set rule, http://www.ncvhs.hhs.gov/091210p06b.pdf.

[58] The European Parliament and the Council. EU Directive 95/46/EC, http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML.

[59] Benny Pinkas. Cryptographic techniques for privacy-preserving data mining. *ACM SIGKDD Explorations Newsletter*, 4(2):12–19, December 2002.

[60] Giorgos Poulis, Aris Gkoulalas-Divanis, Grigorios Loukides, Spiros Skiadopoulos, and Christos Tryfonopoulos. SECRETA: A system for evaluating and comparing relational and transaction anonymization algorithms. In *Proceedings of the 17th International Conference on Extending Database Technology, EDBT Athens, Greece, March 24-28, 2014*, pages 620–623, 2014.

[61] Giorgos Poulis, Grigorios Loukides, Aris Gkoulalas-Divanis, and Spiros Skiadopoulos. Anonymizing data with relational and transaction attributes. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD (3))*, pages 353–369, 2013.

[62] Kris Rogers, Anna Kemp, Andrew McLachlan, and Fiona Blyth. Patterns of prescription opioid use for non-cancer pain in australia: findings from the new south wales 45 and up study. *International Data Linkage Conference*, 2012.

[63] Pierangela Samarati. Protecting respondents identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[64] Pierangela Samarati and Sweeney Latanya. Generalizing data to provide anonymity when disclosing information. In *Proceedings of the Seventeenth (ACM) (SIGACT-SIGMOD-SIGART) Symposium on Principles of Database Systems*, page 188, 1998.

[65] Ravi S Sandhu, Edward J Coyne, Hal L Feinstein, and Charles E Youman. Role-based access control models. *Computer*, 29(2):38–47, Feb 1996.

[66] Jimeng Sun and Chandan K Reddy. Big data analytics for healthcare. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1525–1525. ACM, 2013.

[67] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[68] Tsubasa Takahashi, Koji Sabataka, and Takuya Mori. Efficient and tailor-made anonymization for relational and transactional medical records. In *Workshop on Data Science for Social Good at KDD*, 2014.

[69] Acar Tamersoy, Grigorios Loukides, Mehmet Ercan Nergiz, Yucel Saygin, and Bradley Malin. Anonymization of longitudinal electronic medical records. *Information Technology in Biomedicine, IEEE Transactions on*, 16(3):413–423, 2012.

[70] Manolis Terrovitis, John Liagouris, Nikos Mamoulis, and Spiros Skiadopoulos. Privacy preservation by disassociation. *Proceedings of the VLDB Endowment (PVLDB)*, 5(10):944–955, 2012.

[71] Manolis Terrovitis, Nikis Mamoulis, and Panos Kalnis. Privacy-preserving anonymization of set-valued data. *Proceedings of the VLDB Endowment (PVLDB)*, 1(1), 2008.

[72] Manolis Terrovitis, Nikos Mamoulis, and Panos Kalnis. Local and global recoding methods for anonymizing set-valued data. *The VLDB Journal – The International Journal on Very Large Data Bases*, 20(1):83–106, 2011.

[73] Josep Domingo-Ferrerand Vicenç Torra. Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Mining and Knowledge Discovery*, 11(2):195–212, 2005.

[74] Grigorios Tsoumakas and Ioannis Katakis. Multilabel classification: An overview. *International Journal of Data Warehousing and Mining*, pages 1–13, 2007.

[75] Xiang Wang, David Sontag, and Fei Wang. Unsupervised learning of disease progression models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 85–94. ACM, 2014.

[76] Xiaokui Xiao and Yufei Tao. Anatomy: Simple and effective privacy preservation. In *32nd International Conference on Very Large Data Bases*, VLDB '06, pages 139–150. VLDB Endowment, 2006.

[77] Xiaokui Xiao and Yufei Tao. Personalized privacy preservation. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 229–240. ACM, 2006.

[78] Jian Xu, Wei Wang, Jian Pei, Xiaoyuan Wang, Baile Shi, and Ada Wai chee Fu. Utility-based anonymization using local recoding. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–790, 2006.

[79] Jian Xu, Wei Wang, Jian Pei, Xiaoyuan Wang, Baile Shi, and Ada Wai-Chee Fu. Utility-based anonymization using local recoding. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–790. ACM, 2006.

[80] Yabo Xu, Ke Wang, Ada Wai-Chee Fu, and Philip S. Yu. Anonymizing transaction databases for publication. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 767–775. ACM, 2008.

Appendix A. (k, k^m)-anonymization

In this appendix, we devise a method for (k, k^m)-anonymizing a cluster of records in a way that satisfies the utility constraint set and incurs minimal information loss according to the *UL* measure. Our method is called **GenDiag** and is based on the CBA algorithm [45], discussed in Section 2.2. **GenDiag** differs from CBA in that it enforces k^m-anonymity. The pseudocode of our method is illustrated in Algorithm **GenDiag**. Our method takes as input a cluster of records C, a utility constraint set $\mathcal{U}$, and the anonymization parameters k and m, and it outputs a (k, k^m)-anonymous cluster C' corresponding to C and the number of diagnosis codes s that have been suppressed from C.

Algorithm: GenDiag

Input: A cluster of records C, a utility constraint set $\mathcal{U}$, and the anonymization parameters k and m.
Output: A (k, k^m)-anonymous cluster C' corresponding to C and the number of diagnosis codes s that have been suppressed from C.

```
1  Initialize C' = C and s = 0.
2  Let Q be the set that contains sets of diagnosis codes, each of which contains up to m codes and appears in fewer than
     k records of C'.
3  while Q is not empty do
4      Let p be the element of Q that appears in the largest number of records.
5      while p appears in fewer than k records of C' do
6          Find codes u and u', such that (a) u is contained in p, (b) u and u' are contained in the same utility
             constraint of U, and (c) UL(ũ) is minimum, where ũ is the generalized diagnosis code containing the set of
             diagnosis codes that appear in u or u'.
7          if u and u' are found then
8              Replace u and u' by ũ in p, Q and in all records of C'.
9          else while p appears in fewer than k records of C do
10             Let e be the element of p that appears in the minimum number of records of C.
11             Increase s by the number of diagnosis codes that appear in e.
12             Suppress e from p, Q and all records of C'.
13         ;
14     Remove p from Q.
15 return C' and s.
```

In Step 1, **GenDiag** initializes the output variables C' ans s, and in Step 2, it finds all sets of up to m diagnosis codes that appear in fewer than k records of the cluster and stores them in set Q. Next, it iterates over these sets, in decreasing order of frequency (number of records of the cluster in which these sets appear) and performs generalization and/or suppression (Steps 3–14). Specifically, for a set of diagnosis codes p with element u, a generalized diagnosis code $\tilde{u}$ is created (Steps 6–8). The generalized diagnosis code $\tilde{u}$ contains all the diagnosis codes of u as well as additional codes from the same utility constraint (encoded by code u' in Step 6), so that the utility constraint set $\mathcal{U}$ is satisfied. Additionally, **GenDiag** chooses u and u' so that their generalization $\tilde{u}$ incurs minimal information loss (Step 6). If u and u' are found then the generalization is performed (Step 7–8). Otherwise, **GenDiag** iteratively suppresses, from the element p, the diagnosis code that appears in the minimum number of records in the cluster, and it keeps, in variable s, the summation of suppressed diagnosis codes (Steps 9–12).

Generalization and suppression are performed until the set of codes p appears in at least k records of the cluster, and, after that, p is not considered again (Step 14). When the loop of Steps 3–14 terminates, cluster C' is (k, k^m)-anonymous. Thus, **GenDiag** outputs cluster C' and the total number of suppressed diagnosis codes s (Step 15). Example 6 illustrates the operation of **GenDiag**.

Example 6. *Let us apply* **GenDiag** *to the set of records* {0,1,2,3} *of Figure 1a using the utility constraint set of Table 4 and parameters* $k = 2$ *and* $m = 2$ *(the input set corresponds to the union of clusters* $C_1 = \{0,1\}$ *and* $C_2 = \{2,3\}$ *formed in the example of Section 4.2).*

Initially, the set of diagnosis codes $\{493.2, 053.20\}$*, which is contained in fewer than two records, is added into* Q *(Step 2). Next, the element* $p = \{493.2, 053.20\}$ *is retrieved, the pair* 053.20,053.71*, and the generalized diagnosis code* $(053.20, 053.71)$ *is created (Steps 4–6). To verify, note that* 053.71 *is contained in the same utility constraint with* 053.20 *and that the UL of* $(0.5320, 053.71)$ *is minimum. Note also that* 493.2 *(i.e., the other code in* p*) cannot be generalized with any other diagnosis code, as this would violate the utility constraint set of Table 4.*

Following that, **GenDiag** *replaces all occurrences of the code* 053.20 *in the input set with* $(053.20, 053.71)$*, and* p *becomes* $\{493.2, (053.20,053.71)\}$ *(Step 8). Since* p *now appears in three records, it is removed from* Q *(Steps 14), and* **GenDiag** *returns the anonymized output set (illustrated by the first 4 rows of the dataset in Figure 3).*

Appendix B. Complexity of GenDiag

Let C be the input cluster, β be the maximum number of diagnosis codes contained in any record of the cluster C, and γ be the total number of distinct diagnosis codes in the

records of the cluster C. Let also Q be the set that is comprised of sets of diagnosis codes, each of which contains up to m codes and appears in fewer than k records of C. Last, let p be the element of Q that contains the larger number of diagnosis codes.

The first two steps of **GenDiag** need $\mathcal{O}([|C| \cdot \sum_{i \in \{1,\ldots,m\}} \binom{\beta}{i}] + [|Q| \cdot \log(|Q|)])$ time. The first term in square brackets is the time to construct Q (Step 2), and the second term is the time to sort Q. Sorting is required by the next steps of **GenDiag**, to access the elements of Q in decreasing frequency order. The **while** loop of Steps 3–14 is performed $\mathcal{O}(|Q|)$ times, and each iteration of Steps 5–12 needs $\mathcal{O}(|p|^2 \cdot \gamma + |p| \cdot |C|))$ time. Thus, the **while** loop of Steps 3–14 takes $\mathcal{O}(|Q| \cdot (|p|^2 \cdot \gamma + |p| \cdot |C|))$ time.

Therefore, the time complexity of **GenDiag** is:

$$O(|C| \cdot \sum_{i \in \{1,\ldots,m\}} \binom{\beta}{i} + |Q| \cdot (\log(|Q|) + |p|^2 \cdot \gamma + |p| \cdot |C|))$$

or equivalently:

$$O(|C| \cdot \sum_{i \in \{1,\ldots,m\}} \binom{\beta}{i} + |Q| \cdot (\log(|Q|) + m^2 \cdot \gamma + m \cdot |C|))$$

since $|p|$ is at most m.

Appendix C. Time complexity of ART$_{UC}$

To compute the time complexity of ART$_{UC}$, we compute the complexity of each of its phases. In the following, we denote the size of a given a set S with $|S|$.

Record grouping requires a single pass of the RT-dataset. Thus, it can be performed in $\mathcal{O}(|D|)$ time.

Cluster formation requires $\mathcal{O}(\sum_{G \in \mathcal{G}} |G|^2)$ time, where $\mathcal{G}$ is the set of clusters formed in Steps 1 and 2. For each group $G \in \mathcal{G}$, ART$_{UC}$ compares a random records with all others in the set (Step 8) which can be performed in $\mathcal{O}(|G|^2)$ time. Note that the generalization with Definition 2 is performed in an incrementally fashion. Thus, it can be performed in constant time. Also, the final Steps 15–16 need $\mathcal{O}(|\mathcal{C}|)$ time and do not affect the overall complexity.

Cluster merging Step 17 can be implemented with a multidimensional sorting in $\mathcal{O}(\lambda \cdot |\mathcal{C}| \cdot \log(|\mathcal{C}|))$ time, where λ is the number of dimensions, i.e., the number of demographic

attributes. The **repeat/until** loop of Steps 18–22 is executed at most $\mathcal{O}(|\mathcal{C}|^2)$ times. The time needed to execute Steps 19–22 once is $\mathcal{O}(\mu)$, where $\mathcal{O}(\mu)$ is the cost of executing our adaptation of the CBA algorithm with input the two larger clusters of set $\mathcal{C}$ (see Appendix A for more details). In summary, the **repeat/until** loop can be done in $\mathcal{O}(|\mathcal{C}|^2 \cdot \mu)$ time. Thus in total, the cluster merging step can be performed in $\mathcal{O}(|\mathcal{C}|^2 \cdot \mu)$ time.

(k, k^m)**-anonymization** requires $\mathcal{O}(\sum_{C \in \mathcal{C}} (\text{CBA}(C))$ time.

Therefore, the cost time consuming phases of ART_{UC} are cluster formation and merging, and the time complexity of ART_{UC} is $\mathcal{O}(\sum_{G \in \mathcal{G}} |G|^2 + |\mathcal{C}|^2 \cdot \mu)$.